



HAL
open science

Reconstruction d'images TEP par des méthodes d'optimisation hybrides utilisant un réseau de neurones non supervisé et de l'information anatomique

Alexandre Merasli

► To cite this version:

Alexandre Merasli. Reconstruction d'images TEP par des méthodes d'optimisation hybrides utilisant un réseau de neurones non supervisé et de l'information anatomique. Biologie cellulaire. Nantes Université, 2024. Français. NNT: . tel-04559619v1

HAL Id: tel-04559619

<https://inserm.hal.science/tel-04559619v1>

Submitted on 25 Apr 2024 (v1), last revised 2 May 2024 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Table des matières

Table des figures	13
Liste des tableaux	15
Liste des acronymes	17
Introduction - Contexte - Objectifs de la thèse	19
1 Imagerie TEP	21
1.1 Principe généraux TEP	21
1.1.1 Désintégration radioactive	21
1.1.1.1 Éléments physiques de l'isotope aux rayons gammas	21
1.1.2 Interactions des photons gamma avec la matière	22
1.1.2.1 Effet photoélectrique	22
1.1.2.2 Effet Compton	22
1.1.2.3 Diagramme de prédominance de ces effets	23
1.1.2.4 Modélisation de tous ces phénomènes : l'atténuation	23
1.1.3 Fonctionnement d'un scanner TEP	24
1.1.3.1 Cristaux scintillateur	24
1.1.3.2 Photodétecteurs	25
1.1.3.2.1 Tube Photomultiplicateur (PMT)	25
1.1.3.2.2 PhotoMultiplicateurs au Silicium (SiPM)	25
1.1.3.3 Organisation d'un scanner TEP	26
1.1.3.3.1 Organisation en blocs détecteurs	26
1.1.3.3.2 Organisation en anneau	26
1.1.4 Évènements enregistrés par la machine TEP	27
1.1.4.1 Détections simples	27
1.1.4.2 Coïncidences	27
1.1.4.2.1 Coïncidences vraies	28
1.1.4.2.2 Coïncidences fortuites	28
1.1.4.2.3 Coïncidences diffusées	28
1.1.4.2.4 Coïncidences multiples	29
1.1.4.3 Temps de vol (TOF)	29
1.1.5 Principe d'acquisition	29
1.1.5.1 Format sinogramme	29
1.1.5.1.1 Transformée de Radon	30
1.1.5.1.2 Sinogramme 2D	30
1.1.5.2 Format mode liste	31
1.1.5.3 Données temps de vol	31
1.2 Des données acquises à l'image reconstruite	32
1.2.1 Modèle direct et corrections	32
1.2.1.1 Modèle direct	32
1.2.1.2 Termes de correction	33
1.2.1.2.1 Matrice système	33

1.2.1.2.1.1	Point Spread Function	33
1.2.1.2.1.2	Projecteur géométrique	34
1.2.1.2.1.3	Correction d'atténuation	34
1.2.1.2.1.4	Normalisation	35
1.2.1.2.2	Correction des coïncidences diffusées	35
1.2.1.2.3	Correction des coïncidences fortuites	35
1.2.1.2.4	Facteur de calibration	36
1.2.1.2.5	Autres corrections	36
1.2.2	Le problème inverse de la reconstruction d'images TEP	36
1.2.3	Reconstruction analytique	36
1.2.4	Modélisation de la désintégration radioactive et de sa détection par la machine TEP	37
1.2.5	Reconstruction de données en format sinogramme par méthodes itératives	38
1.2.5.1	Vraisemblance de Poisson	38
1.2.5.2	Problème d'optimisation du maximum de vraisemblance	39
1.2.5.3	Nature mal posée du problème TEP et solutions possibles	40
1.2.5.4	Ajout d'un terme de régularisation à la vraisemblance	40
1.2.5.5	Méthodes itératives cliniques	41
1.2.5.5.1	Algorithme MLEM	41
1.2.5.5.2	Algorithme OSEM	42
1.2.5.5.3	Algorithme BSREM	43
1.2.5.5.4	Contrainte intrinsèque de non-négativité	44
1.2.5.6	Méthodes itératives pour réduction de biais	45
1.2.5.6.2	Algorithme AML	46
1.2.5.6.3	Algorithme APPGML	46
1.2.5.6.4	Algorithme ADMM-Reg	46
1.2.5.7	Fonctions de régularisation	47
1.2.5.7.1	Pénalité quadratique	47
1.2.5.7.2	Pénalité MRF	48
1.2.5.7.2.1	Différentes fonctions potentielles	48
1.2.5.7.2.2	Pénalité RDP	48
1.2.5.7.2.3	Méthode de Bowsher	49
1.2.6	Méthodes de reconstruction avec réseau de neurones	50
1.2.6.1	Méthodes supervisées	50
1.2.6.1.1	Reconstruction directe avec réseau de neurones	50
1.2.6.1.2	Méthodes de unrolling	50
1.2.6.1.3	ADMM Plug and Play	50
1.2.6.2	Méthodes non supervisées	50
1.3	Applications cliniques d'intérêt pour cette thèse	50
1.3.1	Machines TEP utilisées en clinique	50
1.3.2	Applications au [18F]-FDG	51
1.3.3	Applications en radioembolisation à l' ^{90}Y	51
1.3.3.1	L'Yttrium-90 (^{90}Y)	51
1.3.3.2	Radioembolisation	51
2	Développement de l'algorithme DNA	53
2.1	Introduction	53
2.2	Contexte	54
2.2.1	Réseau de neurones	54
2.2.1.1	Entraînement d'un réseau de neurones	54
2.2.1.2	Architecture d'un réseau de neurones	55
2.2.2	Deep Image Prior (DIP)	56
2.2.3	Algorithme DIPRecon	57
2.3	Développement de l'algorithme DNA	58
2.3.1	Dérivation mathématique	58

2.3.2	Méthodes pour résoudre chaque équation de l'algorithme DNA	59
2.3.2.1	Mise à jour de x	61
2.3.2.2	Mise à jour de v	61
2.3.2.3	Mise à jour automatique du paramètre de pénalité d'ADMM ρ_2	62
2.3.2.4	Débruitage DIP à l'intérieur de la reconstruction	62
2.3.3	Architecture du réseau de neurones utilisée pour le DIP	62
2.3.4	Initialisation des algorithmes DNA et DIPRecon	63
2.4	Expériences	63
2.4.1	Expérience basique "DIP ReLU"	64
2.4.2	Simulations TEP 2D	64
2.4.2.1	Fantôme cérébral et simulations analytiques	64
2.4.2.2	Algorithmes de reconstruction	65
2.4.2.3	Métriques d'évaluation	66
2.4.3	Données réelles TEP 3D	67
2.5	Résultats	68
2.5.1	Expérience DIP ReLU	68
2.5.2	Simulations 2D TEP	69
2.5.3	Données réelles 3D TEP	71
2.6	Discussion	71
2.7	Reconstruction sur des données ^{90}Y	74
2.7.1	Introduction du DNA dans le contexte de la radioembolisation à l' ^{90}Y	74
2.7.2	Simulations 2D ^{90}Y et paramètres des algorithmes de reconstruction	74
2.7.2.1	Fantôme ^{90}Y	74
2.7.2.2	Algorithmes de reconstruction	75
2.7.3	Résultats sur le fantôme ^{90}Y	75
2.7.4	Discussion sur le fantôme ^{90}Y	77
2.8	Conclusion	79
3	Étude et optimisation des hyperparamètres du DNA	81
3.1	Introduction	81
3.2	Ensemble des hyperparamètres du DNA	81
3.2.1	Fantômes et simulations analytiques	82
3.2.1.1	Fantômes	82
3.2.1.1.1	Fantôme cylindrique 1	82
3.2.1.1.2	Fantôme cylindrique 3	83
3.2.1.2	Simulations analytiques	83
3.3	Hyperparamètres liés à l'initialisation utilisée pour le DNA	83
3.3.1	Choix de l'image de départ	83
3.3.1.1	Expériences	84
3.3.1.2	Résultats	84
3.3.1.3	Discussion	86
3.3.1.4	Conclusion	86
3.3.2	Méthodes d'arrêt précoce pour le DIP	86
3.3.2.1	Intuition pour les méthodes d'arrêt précoce pour le DIP	86
3.3.2.2	Calcul de la variance mobile	87
3.3.2.2.1	Méthode de la WMV	87
3.3.2.2.2	Méthode de l'EMV	87
3.3.2.2.3	Implémentation numérique	88
3.3.2.3	Expériences	88
3.3.2.4	Résultats	88
3.3.2.5	Discussion	90
3.3.2.6	Conclusion	90
3.3.3	Impact de la ReLU finale dans un débruitage avec un réseau DIP	90
3.3.3.1	Expériences	90
3.3.3.2	Résultats	90

3.3.3.3	Discussion	90
3.3.3.4	Conclusion	92
3.4	Hyperparamètres liés à la contrainte de non-négativité	92
3.4.1	Équilibre des résidus relatifs pour la mise à jour automatique du paramètre ρ_2	92
3.4.1.1	Méthodes	92
3.4.1.2	Conclusion	94
3.4.2	Remarques sur le choix de l'algorithme interne optimisant la vraisemblance	94
3.4.2.1	Comparaison de ADMM-Reg et APPGML	94
3.4.2.1.1	Contexte	94
3.4.2.1.2	Expériences	94
3.4.2.1.3	Résultats	94
3.4.2.1.4	Discussion	96
3.4.2.2	Remplacement de l'algorithme ADMM-Reg par APPGML à l'intérieur du DNA	96
3.4.2.3	Méthodes	96
3.4.2.3.1	Expériences	97
3.4.2.3.2	Résultats	97
3.4.2.3.3	Discussion	99
3.4.2.3.4	Conclusion	100
3.5	Hyperparamètres liés au DIP au sein de la reconstruction	100
3.5.1	Étude de l'optimiseur du DIP	100
3.5.1.1	Optimiseurs utilisés en TEP avec un réseau DIP	100
3.5.1.1.1	Adam	100
3.5.1.1.2	L-BFGS	101
3.5.1.2	Expériences	101
3.5.1.3	Résultats	103
3.5.1.4	Discussion	103
3.5.1.5	Conclusion	103
3.5.2	Méthodes de mise à l'échelle pour le DIP	103
3.5.2.1	Expériences	103
3.5.2.2	Résultats	104
3.5.2.3	Discussion	104
3.5.2.4	Conclusion	106
3.5.3	Influence du paramètre ρ_1	106
3.5.3.1	Expériences	106
3.5.3.2	Résultats	106
3.5.3.3	Discussion	108
3.5.3.4	Conclusion	109
3.5.4	Influence du taux d'apprentissage	109
3.5.4.1	Correspondance en bande de fréquence (FBC)	109
3.5.4.2	Expériences	110
3.5.4.3	Résultats	111
3.5.4.4	Discussion	113
3.5.4.5	Conclusion	113
3.5.5	Étude statistique sur les sources de variabilité intervenant dans un débruitage DIP	113
3.5.5.1	Expériences	114
3.5.5.1.1	Expérience 1	114
3.5.5.1.2	Expérience 2	114
3.5.5.2	Résultats	114
3.5.5.3	Discussion	114
3.5.5.4	Conclusion	115
3.5.6	Étude sur l'architecture du DIP	115

3.5.6.1	Contexte	115
3.5.6.2	Méthodologie générale	117
3.5.6.3	Étude sur l'architecture d'un réseau convolutionnel pour le DIP	117
3.5.6.3.1	Expériences	117
3.5.6.3.2	Résultats	117
3.5.6.3.3	Discussion	119
3.5.6.4	Ajout de couches de contrôle spectral	119
3.5.6.4.1	Expériences	119
3.5.6.4.2	Résultats	120
3.5.6.4.3	Discussion	120
3.5.6.5	Ajouter du bruit à l'entrée ou à l'espace latent du réseau DIP	120
3.5.6.5.1	Expériences	120
3.5.6.5.2	Résultats	121
3.5.6.5.3	Discussion	122
3.5.6.6	Architecture utilisant les transformateurs	122
3.5.6.6.1	Expériences	123
3.5.6.6.2	Résultats	123
3.5.6.6.3	Discussion	125
3.5.6.7	Discussion générale sur l'architecture du DIP en fonction de l'image d'entrée	125
3.6	Choix du nombre d'itérations des blocs itératifs du DNA	126
3.6.1	Influence du nombre d'itérations du DIP au sein de la reconstruction	126
3.6.1.1	Expériences	126
3.6.1.2	Résultats	126
3.6.1.3	Discussion	128
3.6.1.4	Conclusion	128
3.6.2	Choix du nombre de sous-itérations internes	128
3.6.3	Influence du nombre d'itérations internes	128
3.6.4	Choix du nombre d'itérations externes	129
3.6.4.1	Observations	129
3.6.4.2	Perspectives	130
3.6.4.2.1	Extension de l'algorithme EMV	130
3.6.4.2.2	Pallier le surapprentissage du DIP dans le cadre d'une intégration à la reconstruction TEP	130
3.6.4.3	Conclusion	131
3.7	Discussion générale sur les hyperparamètres et conclusion	131
4	Influence de l'entrée du DIP et des sauts de connexion	133
4.1	Introduction	133
4.2	Simulations analytiques et paramètres de reconstruction généraux	134
4.2.1	Fantôme 2D cérébral et simulations	134
4.2.2	Paramètres de reconstruction	135
4.3	Expériences préliminaires	135
4.4	Méthodes proposées pour mitiger les discordances	137
4.4.1	Méthode "MR init"	137
4.4.2	Méthode "MR 5"	137
4.5	Évaluation des méthodes proposées	139
4.5.1	Expériences	139
4.5.2	Résultats	139
4.5.2.1	Résultats avec la méthode "MR init"	139
4.5.2.2	Résultats avec la méthode "MR 5"	141
4.5.2.3	Comparaison des deux méthodes proposées	144
4.6	Discussion	146
4.7	Conclusion	147
	Discussion générale - Conclusion - Perspectives	149

Bibliographie	151
Annexes	161
A	Liste des publications et présentations scientifiques 161
A.1	Liste des publications scientifiques 161
B	Optimisation mathématique 161
B.1	Algorithme ADMM 161
B.1.1	Lagrangien 161
B.1.2	Algorithme de montée de gradient duale 162
B.1.3	Lagrangien augmenté 163
B.1.4	ADMM 163
B.1.5	Résidus associés au problème d'optimisation 164
B.2	Calcul de pas analytique pour la mise à jour de x dans l'algorithme DNA 165
B.3	Dérivation plus complète de l'algorithme ADMM-Reg 165
B.3.1	Problème d'optimisation initial 165
B.3.2	Détails sur la mise à jour du sinogramme v 166
B.3.3	Discussion sur l'article de Lim et al. 168
C	Outils numériques 169
C.1	CASToR 169
C.1.1	Présentation détaillée du fonctionnement de CASToR 169
C.1.2	Contributions apportées au logiciel CASToR 169
D	Résultats additionnels 170
D.1	Autres résultats sur l'expérience basique "DIP ReLU" 170

Table des figures

1.1	Effet photoélectrique	22
1.2	Effet Compton.	22
1.3	Interactions de photons avec la matière en fonction de l'énergie du photon et du numéro atomique de la matière absorbante.	23
1.4	Facteur d'atténuation μ pour modéliser le nombre de photons traversant une paroi d'épaisseur x sans interagir avec elle.	24
1.5	Organisation d'un bloc détecteur.	26
1.6	Différents axes et plans à travers le corps humain. En particulier, la direction axiale est perpendiculaire au plan transverse (source : www.wikipedia.org).	26
1.7	Blocs détecteurs et anneaux du scanner TEP Siemens Biograph Vision. La partie de gauche de la figure montre l'agencement des blocs détecteurs en anneau, ainsi que plusieurs anneaux de détection juxtaposés selon la direction axiale z . La partie droite de cette figure montre un bloc détecteur formés de plusieurs matrices de cristaux (source : https://radiology.weill.cornell.edu/research/research-projects/long-axial-field-view-pet).	27
1.8	Fenêtre temporelle ouverte à chaque détection simple	27
1.9	Coïncidence fortuite.	28
1.10	Coïncidence diffusée.	28
1.11	Transformée de Radon $R(x(r, \phi))$ pour un point de radioactivité dans l'espace de coordonnées (u, v)	29
1.12	Sinogramme 2D pour un point de radioactivité. Seules les valeurs le long d'une ligne sinusoïdale blanche sont non nulles.	31
1.13	Probabilité d'une annihilation sur une LOR sans information TOF par rapport à une LOR avec information TOF.	32
1.14	Deux objets circulaires vue de façon floue par le système TEP, modélisé par la PSF.	34
1.15	Convergence d'algorithme utilisant des sous-ensembles (comme OSEM) par rapport au même algorithme n'en utilisant pas (comme MLEM).	43
1.16	Injection pour un traitement par radioembolisation.	52
2.1	Phénomène d'apprentissage précoce puis surapprentissage du DIP.	56
2.2	Ensemble des équations de l'algorithme DNA. Chaque bloc de ce diagramme est itératif (cf. partie 2.3.2).	60
2.3	Architecture du DIP utilisée pour le DNA. Légende : Conv = couche convolutionnelle; BN = Batch Normalization; LReLU = LeakyReLU; Conv stride2 = couche convolutionnelle avec stride 2.	62
2.4	Fantôme conçu pour mettre en évidence les artefacts causés par la ReLU (<i>gauche</i> : GT TEP avec insert froid; <i>droite</i> : GT TEP avec insert chaud).	64
2.5	Fantôme cérébral utilisé pour les simulations TEP FDG 2D (<i>gauche</i> : image IRM pondérée T1; <i>milieu</i> : GT TEP); <i>droite</i> : image d'atténuation en cm^{-1}	64

2.6	Expérience DIP ReLU : reproduire une image TEP (<i>première ligne à droite</i>) avec le DIP prenant en entrée une image anatomique (<i>première ligne à gauche</i>). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (<i>colonne de gauche</i>) et sans ReLU (<i>colonne de droite</i>), après 1000 itérations (<i>deuxième ligne</i>) et 9000 itérations (<i>troisième ligne</i>), lors de l’ajustement de l’image TEP avec l’insert froid. L’image anatomique contient une zone en hyposignal.	68
2.7	Courbes de compromis moyennées sur 20 réplicats de la récupération d’activité dans la ROI tumeur (<i>haut gauche</i>), du biais relatif dans la ROI LCR (<i>haut droite</i>), du biais relatif dans la ROI matière blanche (<i>bas gauche</i>) et du SSIM dans l’ensemble du cerveau (<i>bas droite</i>), en fonction de la rugosité dans l’ensemble de la matière blanche, pour l’algorithme DNA proposé, DIPRecon, BSREM + RDP avec et sans pondérations de Bowsher, OSEM, APPGML et ADMM-Reg avec pénalité MRF quadratique et pondérations de Bowsher. Le BSREM de départ utilisé pour DIPRecon et DNA correspond à la croix noire. Les diamants blancs correspondent à cette image débruitée par le DIP (avec et sans ReLU) à l’aide de l’algorithme d’arrêt précoce EMV.	69
2.8	Images entre 10 et 25 % de rugosité pour tous les algorithmes calculés. L’image BSREM de départ et les initialisations DIP-BSREM utilisées pour initialiser le DIPRecon et le DNA sont aussi présentées.	70
2.9	Coupe axiale des images reconstruites à partir des données réelles 3D. L’image IRM T1 est représentée sur la <i>première ligne</i> . L’image BSREM clinique (avec et sans les poids de Bowsher) ainsi que l’image de départ BSREM sont représentées sur la <i>deuxième ligne</i> . Les initialisations DIP-BSREM utilisées pour initialiser le DIPRecon et le DNA, ainsi que les images reconstruites par DIPRecon et DNA sont indiquées sur la <i>dernière ligne</i>	72
2.10	Fantôme utilisé pour les simulations TEP 2D ^{90}Y (<i>gauche</i> : image anatomique; <i>milieu</i> : GT TEP); <i>droite</i> : image d’atténuation en cm^{-1}	74
2.11	Courbes de compromis moyennées sur 20 réplicats pour la récupération d’activité dans la ROI tumeur “TEP seule” (<i>haut gauche</i>) et la ROI tumeur “TEP match” (<i>haut droite</i>), le biais relatif dans la ROI de la zone froide (<i>bas gauche</i>) et le SSIM dans l’ensemble du fantôme (<i>bas droite</i>), en fonction de la rugosité dans le fond du fantôme, pour l’algorithme DNA proposé, DIPRecon, BSREM+RDP avec et sans poids de Bowsher, OSEM, et APPGML et ADMM-Reg avec pénalité MRF quadratique. Le BSREM de départ utilisé pour DIPRecon et DNA correspond à la croix noire. Les diamants blancs correspondent à cette image BSREM de départ débruitée par le DIP à l’aide de l’algorithme d’arrêt précoce EMV.	76
2.12	Images reconstruites du fantôme ^{90}Y pour tous les algorithmes étudiés. L’image BSREM de départ et l’initialisation DIP-BSREM utilisée pour initialiser le DNA est aussi présentée. L’image OSEM présentée ici est montrée après 2 itérations et 28 sous-ensembles, avec un post filtrage gaussien avec FWHM de 4mm.	78
3.1	Liste des hyperparamètres liés à l’optimisation du DNA tel que présenté dans le chapitre précédent.	82
3.2	Fantôme cylindrique 1. A gauche, la TEP, à droite l’image d’atténuation en cm^{-1} utilisée aussi comme entrée du DIP.	82
3.3	Fantôme cylindrique 3. A gauche l’image anatomique utilisée en entrée du DIP, au milieu la GT TEP, à droite l’image d’atténuation en cm^{-1}	83
3.4	Courbes de compromis moyennées sur 100 réplicats sur le fantôme cylindrique 1 dans la ROI de la tumeur (gauche) et la ROI de la zone froide (droite), pour le DNA et le DIPRecon, en faisant varier l’image de départ (ADMM-Reg 100 itérations et BSREM clinique). Les diamants blancs correspondent à l’initialisation DIP-BSREM pour chaque algorithme.	84

3.5	Courbes de compromis moyennées sur 20 réplicats sur le fantôme cylindrique 2 dans la ROI de la tumeur “TEP seule” (haut gauche), la ROI de la tumeur “TEP match” (haut droite), la ROI de la zone froide (bas gauche) et le SSIM dans l’ensemble du fantôme (bas droite), pour le DNA et le DIPRecon, en faisant varier l’image de départ (ADMM-Reg 300 itérations, MLEM 60 itérations et BSREM clinique). Les diamants blancs correspondent à l’initialisation DIP-BSREM pour chaque algorithme.	85
3.6	Principe de la variance mobile pour pratiquer un arrêt précoce pour le DIP . . .	87
3.7	MSE, SSIM et EMV (échelle logarithmique) pour un débruitage DIP en fonction des itérations du DIP. Les 200 premières itérations ont été retirées pour une meilleure lisibilité. La ligne verticale verte indique l’itération trouvée par l’algorithme EMV.	89
3.8	Images obtenues après un débruitage DIP de l’image de départ BSREM présentée à gauche. La deuxième image est obtenue à l’itération trouvée par l’algorithme EMV. La troisième correspond à l’image obtenue après 600 itérations du DIP (correspondant environ au minimum de MSE). L’image de droite correspond à l’image obtenue après 800 itérations du DIP (correspondant environ au maximum de SSIM).	89
3.9	Courbes de compromis moyennées sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur (<i>haut gauche</i>), la ROI du LCR (<i>haut droite</i>), la ROI de la matière blanche (<i>bas gauche</i>) et le SSIM dans l’ensemble du cerveau (<i>bas droite</i>). Les courbes montrent l’évolution au cours des itérations des métriques sur les initialisations DIP-BSREM utilisées pour le DNA et le DIPRecon. Les 100 premières itérations ont été retirées du graphe pour la lisibilité.	91
3.10	Récupération d’activité au cours des itérations du DIP, moyennée sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur pour les initialisations DIP-BSREM utilisées pour le DNA et le DIPRecon. Les 100 premières itérations ont été retirées du graphe pour la lisibilité.	91
3.11	Courbes de compromis moyennées sur 20 réplicats pour le fantôme cylindrique 2 en situation ^{90}Y , dans la ROI de la tumeur “TEP seule” (<i>haut gauche</i>), la ROI de la tumeur “TEP match” (<i>haut droite</i>), la ROI de la zone froide (<i>bas gauche</i>) et le SSIM dans l’ensemble du fantôme (<i>bas droite</i>), pour les algorithmes APPGML et ADMM-Reg avec pénalité MRF quadratique. Plusieurs valeurs de bornes A sont présentées pour APPGML.	95
3.12	Courbes de compromis moyennées sur 20 réplicats pour le fantôme cylindrique 3 en situation ^{90}Y , dans la ROI de la tumeur “TEP seule” (<i>haut gauche</i>), la ROI de la tumeur “TEP match” (<i>haut droite</i>), la ROI de la zone froide (<i>bas</i>)), pour l’algorithme DNA proposé et la version modifiée DNA-APPGML. Le nombre d’itérations internes des deux algorithmes a été varié. Les diamants blancs correspondent à la première itération externe pour tous les algorithmes étudiés.	97
3.13	Courbes de compromis moyennées sur 20 réplicats pour le fantôme cérébral FDG dans la ROI de la tumeur (<i>haut gauche</i>), la ROI du LCR (<i>haut droite</i>), la ROI de la matière blanche (<i>bas gauche</i>) et le SSIM dans l’ensemble du cerveau (<i>bas droite</i>), pour l’algorithme DNA proposé et la version modifiée DNA-APPGML. Le nombre d’itérations internes des deux algorithmes a été varié. Les diamants blancs correspondent à la première itération externe pour tous les algorithmes étudiés.	98
3.14	Images obtenues pendant la reconstruction du fantôme cylindrique 3 avec l’algorithme DNA-APPGML et une borne $A = -10$. La première ligne montre les images au bout de 2 et 3 itérations externes du DNA-APPGML. La deuxième ligne montre les images obtenues pendant la troisième itération externe lors de la reconstruction par APPGML au sein du DNA-APPGML (à 5 et 6 itérations internes).	99

3.15	Courbes de compromis moyennées sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur (<i>haut gauche</i>), la ROI du LCR (<i>haut droite</i>), la ROI de la matière blanche (<i>bas gauche</i>) et le SSIM dans l'ensemble du cerveau (<i>bas droite</i>). Les courbes montrent l'évolution des métriques au cours des itérations en fonction de l'optimiseur choisi. Les 100 premières itérations de Adam et respectivement les 30,6,1 premières itérations de L-BFGS ont été retirées du graphe pour la lisibilité (respectivement pour valeurs de taux d'apprentissage 0,01,0,1,1).	102
3.16	Courbes de compromis moyennées sur 15 réplicats sur le fantôme cérébral FDG dans la ROI de la tumeur (<i>haut gauche</i>), la ROI du LCR (<i>haut droite</i>), la ROI de la matière blanche (<i>bas gauche</i>) et le SSIM dans l'ensemble du cerveau (<i>bas droite</i>), pour l'algorithme DNA proposé et le DIPRecon, en faisant varier la méthode de mise à l'échelle (standardisation, normalisation ou normalisation au maximum, repérées par les abréviations stand, norm et positive norm sur le graphe). Les diamants blancs correspondent à l'initialisation DIP-BSREM pour chaque algorithme. Certains réplicats du DNA et du DIPRecon ne sont pas montrés car l'algorithme EMV n'a pas détecté de minimum.	105
3.17	Courbes de compromis pour 1 réplicat pour le fantôme cérébral FDG, dans la ROI de la tumeur (<i>haut gauche</i>), la ROI du LCR (<i>haut droite</i>), la ROI de la matière blanche (<i>bas gauche</i>) et le SSIM dans l'ensemble du cerveau (<i>bas droite</i>). L'algorithme DNA-APPGML a été utilisé avec 10 itération internes et différentes valeurs de ρ_1 .	107
3.18	Courbes de compromis pour 1 réplicat pour le fantôme cérébral FDG, dans la ROI de la tumeur. L'algorithme DNA-APPGML a été utilisé avec 1, 10 ou 30 itérations internes (respectivement à gauche, au milieu et à droite). Les axes des abscisses ne sont pas à la même échelle.	107
3.19	Images du fantôme cérébral du chapitre 2 reconstruites avec le DNA, pour $\rho_1 = 3$ et 10 itérations internes. L'image de <i>gauche</i> est reconstruite après 3 itérations externes et celle de <i>droite</i> après 5 itérations externes.	108
3.20	Image du fantôme cylindrique 2 reconstruite après 1000 itérations avec l'algorithme ADMM-Reg sans régularisation.	109
3.21	Masques appliqués sur la transformée de Fourier d'une image. Ces masques sont utilisés dans le calcul de la FBC.	110
3.22	FBC des images en sortie du DIP en fonction des itérations. L'axe des ordonnées représente la FBC entre 0 et 1. L'axe des abscisses représente les itérations de 0 à 5000. Chaque sous figure correspond aux résultats obtenus pour un taux d'apprentissage γ différent.	111
3.23	Images produites par le réseau DIP pour différentes valeurs de taux d'apprentissage.	112
3.24	Boîtes à moustaches représentant la distribution du MSE peak pour l'expérience 1 : 100 réplicats reconstruits par BSREM+RDP et débruités par le DIP.	115
3.25	Boîtes à moustaches représentant la distribution du MSE peak pour l'expérience 2 : 100 entrées différentes du réseau DIP (<i>première ligne</i>), 100 initialisations des poids du réseau DIP pour différentes méthodes (<i>deuxième ligne</i>), 100 débruitages avec entrées et poids qui varient simultanément (<i>troisième ligne</i>).	116
3.26	Boîtes à moustaches de la distribution du MSE peak pour différentes valeurs de profondeur et nombre de canaux pour les architectures "our w/ SC" (<i>gauche</i>), "our Decoder" (<i>milieu</i>) et DD (<i>droite</i>).	118
3.27	Images obtenues pour les trois architectures étudiées pour le débruitage DIP. La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour chaque architecture.	118
3.28	Images obtenues pour les quatre architectures étudiées pour le débruitage DIP, avec ou sans contrôle LG pour les architectures avec uniquement un décodeur. La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour chaque architecture.	120
3.29	Boîtes à moustaches pour les deux expériences avec injection de bruit dans l'image d'entrée (v_0) ou dans l'espace latent (v_1), pour différentes valeurs de σ_p .	121

3.30 Images obtenues pour l'expérience d'injection de bruit en entrée du réseau DIP, dans le cas v_0 . La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour différentes valeurs de σ_p 121

3.31 Images correspondant au minimum de MSE obtenu pour le fantôme cérébral FDG avec l'architecture "our w/ SC" utilisée en tâche de débruitage DIP, pour différents nombres de SC. 123

3.32 Boîtes à moustaches obtenues avec l'architecture "our w/ SC" utilisée en tâche de débruitage DIP pour différents nombres de SC. 124

3.33 Images correspondant au minimum de MSE obtenue pour le fantôme cérébral FDG avec les architectures "our w/ SC", Swin Unet et Swin UNETR utilisées en tâche de débruitage DIP. 124

3.34 Courbes de compromis pour 15 réplicats pour le fantôme cylindrique 2, dans la ROI de la tumeur "TEP seule" (*haut gauche*), la ROI de la tumeur "TEP match" (*haut droite*), la ROI de la zone froide (*bas gauche*) et le SSIM dans l'ensemble du fantôme (*bas droite*) en fonction de la rugosité dans le fond du fantôme, pour l'algorithme DNA utilisant 30, 100 ou 300 itérations du DIP. Les diamants blancs montrent les métriques à la première itération externe. 127

3.35 Valeur de l'EMV en fonction des itérations du DIP lors d'une reconstruction avec l'algorithme DNA utilisant l'EMV à chaque itération externe. Chaque courbe correspond à l'EMV calculée pour une itération externe donnée. L'itération -1 correspond à l'initialisation du DNA. Les courbes s'arrêtent 200 itérations après l'itération choisie par l'algorithme EMV. 127

3.36 Images reconstruites par le DNA avec utilisation de l'algorithme EMV à chaque itération du DIP au sein de la reconstruction. L'image de gauche montre l'image reconstruite par le DNA après 46 itérations externes, celle de droite après 47 itérations externes. 128

3.37 Images reconstruites des simulations ^{90}Y du fantôme cylindrique 2 avec l'algorithme DNA (même paramétrage qu'en partie 2.7.2.2). L'initialisation DIP-BSREM du DNA est présentée en haut à gauche, puis chaque image correspond à un nombre d'itérations externes de 100 jusqu'à 600 et la dernière image est pour 1700. 129

4.1 Fantôme cérébral conçu pour l'étude des mismatches entre image TEP et image anatomique. À gauche, l'image anatomique; au milieu, la vérité de base de la TEP; à droite, l'image d'atténuation en cm^{-1} . Les cadres montrent les 3 types de tumeurs considérées : "TEP match" (vert), "TEP match" (bleu) et "IRM seule" (rouge). 134

4.2 Images du fantôme cérébral FDG reconstruites avec l'algorithme DNA. La première ligne montre la GT (*première image*) et différentes reconstructions : en utilisant une image aléatoire en entrée du réseau DIP et 3 SC (*deuxième image*), une image IRM en entrée du réseau DIP et 3 SC (*troisième image*), et une image proche de la routine clinique reconstruite par BSREM (*quatrième image*). La deuxième ligne montre un zoom pour ces trois images sur la zone "IRM seule". 136

4.3 Images du fantôme cérébral FDG reconstruites avec l'algorithme DNA. La première ligne montre les reconstructions en utilisant une image aléatoire en entrée du réseau DIP et 0 SC (*gauche*) et une image IRM en entrée du réseau DIP et 0 SC (*droite*). La deuxième ligne montre un zoom pour ces deux images sur la zone "IRM seule". 137

4.4 Images reconstruites par le DNA-APPGML en utilisant la méthode "MR init" pour différents nombre de SC (*deuxième et troisième images*), en comparaison avec le DIP conditionnel ("MR 3 SC" *gauche*) et une entrée aléatoire du réseau DIP et 0 SC ("random 0 SC" *droite*). 140

4.5	Courbes de compromis moyennées sur 40 réplicats de la récupération d'activité dans la ROI tumeur match (<i>haut gauche</i>) et la ROI tumeur mismatch (<i>haut droite</i>), du biais relatif dans la ROI "IRM seule" (<i>bas gauche</i>) et du biais relatif dans la ROI LCR (<i>bas droite</i>), en fonction de la rugosité dans l'ensemble de la matière blanche, pour l'algorithme DNA-APPGML proposé. La méthode "MR init" a été utilisée avec 0, 1, 2 ou 3 SC, et a été comparée à l'image IRM en entrée du DIP et 3 SC ainsi qu'à une entrée aléatoire avec 0 SC. Les diamants blancs correspondent à l'initialisation du DNA-APPGML.	140
4.6	Profils le long d'une ligne horizontale au centre de la ROI "IRM seule" montrant le biais relatif de chaque voxel de l'image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode "MR 3 SC" (<i>gauche</i>) et méthode "MR init 2 SC" (<i>droite</i>).	141
4.7	Profil le long d'une ligne horizontale au centre de la ROI "TEP mismatch" montrant le biais relatif de chaque voxel de l'image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode "MR 3 SC" (<i>gauche</i>) et méthode "MR init 2 SC" (<i>droite</i>).	142
4.8	Images reconstruites par le DNA-APPGML en utilisant la méthode "MR 5" (<i>milieu</i>), en comparaison avec le DIP conditionnel ("MR 3 SC" <i>gauche</i>) et une entrée aléatoire du réseau DIP et 0 SC ("random 0 SC" <i>droite</i>).	142
4.9	Courbes de compromis moyennées sur 40 réplicats de la récupération d'activité dans la ROI tumeur match (<i>haut gauche</i>) et la ROI tumeur mismatch (<i>haut droite</i>), du biais relatif dans la ROI "IRM seule" (<i>bas gauche</i>) et du biais relatif dans la ROI LCR (<i>bas droite</i>), en fonction de la rugosité dans l'ensemble de la matière blanche, pour l'algorithme DNA-APPGML proposé. La méthode "MR 5" a été comparée à l'image IRM en entrée du DIP et 3 SC ainsi qu'à une entrée aléatoire avec 0 SC. Les diamants blancs correspondent à l'initialisation du DNA-APPGML.	143
4.10	Profil le long d'une ligne horizontale au centre de la ROI "IRM seule" montrant le biais relatif de chaque voxel de l'image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode "MR 3 SC" (<i>gauche</i>) et méthode "MR 5" (<i>droite</i>).	144
4.11	Profil le long d'une ligne horizontale au centre de la ROI "TEP mismatch" montrant le biais relatif de chaque voxel de l'image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode "MR 3 SC" (<i>gauche</i>) et méthode "MR 5" (<i>droite</i>).	144
4.12	Courbes de compromis moyennées sur 40 réplicats de la récupération d'activité dans la ROI tumeur match (<i>haut gauche</i>) et la ROI tumeur mismatch (<i>haut droite</i>), du biais relatif dans la ROI "IRM seule" (<i>bas gauche</i>) et du biais relatif dans la ROI LCR (<i>bas droite</i>), en fonction de la rugosité dans l'ensemble de la matière blanche, pour l'algorithme DNA-APPGML. La méthode "MR init" a été utilisée avec 2 SC et est comparée à la méthode "MR 5" et à BSREM+RDP avec différentes valeurs de pénalité. Le BSREM de départ utilisé pour DNA-APPGML correspond à la croix noire. Les diamants blancs correspondent à l'initialisation du DNA-APPGML.	145
4.13	SSIM dans l'ensemble du cerveau en fonction de la rugosité de l'image dans l'ensemble de la matière blanche, moyenné sur 40 réplicats, pour l'algorithme DNA-APPGML proposé. La méthode "MR init" a été utilisée avec 2 SC et est comparée à la méthode "MR 5", ainsi qu'aux méthodes avec image IRM en entrée et 3 SC ("MR 3 SC") et une entrée aléatoire avec 0 SC ("random 0 SC"). Les diamants blancs correspondent à l'initialisation du DNA-APPGML.	146

D.1 Expérience DIP ReLU : reproduire une image TEP (*première ligne à droite*) avec le DIP prenant en entrée une image anatomique (*première ligne à gauche*). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (*colonne de gauche*) et sans ReLU (*colonne de droite*), après 1000 itérations (*deuxième ligne*) et 9000 itérations (*troisième ligne*), lors de l'ajustement de l'image TEP avec l'insert chaud. L'image anatomique contient une zone en hypersignal. 170

D.2 Expérience DIP ReLU : reproduire une image TEP (*première ligne à droite*) avec le DIP prenant en entrée une image anatomique (*première ligne à gauche*). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (*colonne de gauche*) et sans ReLU (*colonne de droite*), après 1000 itérations (*deuxième ligne*) et 9000 itérations (*troisième ligne*), lors de l'ajustement de l'image TEP avec l'insert froid. L'image anatomique contient une zone en hypersignal. 171

D.3 Expérience DIP ReLU : reproduire une image TEP (*première ligne à droite*) avec le DIP prenant en entrée une image anatomique (*première ligne à gauche*). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (*colonne de gauche*) et sans ReLU (*colonne de droite*), après 1000 itérations (*deuxième ligne*) et 9000 itérations (*troisième ligne*), lors de l'ajustement de l'image TEP avec l'insert chaud. L'image anatomique contient une zone en hypersignal. 172

Liste des tableaux

1.1	Isotopes utilisés en TEP au CHU de Nantes et implémentés dans le logiciel de reconstruction CASToR	21
1.2	Différents matériaux scintillateurs utilisés en TEP.	25
1.3	Tableau récapitulatif des propriétés probabilistes des désintégrations radioactives	38
3.1	Avantages et inconvénients du DNA par rapport au DNA-APPGML.	100
4.1	Les 5 images utilisées en entrée du réseau DIP avec la méthode “MR 5” et un paramètre $\Sigma = 0.1$	138

Liste des acronymes

- ABML** "A"-B" Maximum Likelihood p.46
- ACF** Attenuation Correction Factor - Facteur de correction d'atténuation p.34
- AML** "A" Maximum Likelihood p.46
- BSREM** Block Sequential Regularized Expectation Maximization p.43
- CASToR** Customizable and Advanced Software for Tomographic Reconstruction - logiciel de reconstruction p.169
- DIP** Deep Image Prior p.56
- DIPRecon** Algorithme de reconstruction utilisant du Deep Image Prior : algorithme DIPRecon p.57
- DNA** DIP Nested ADMM p.58
- ELTO** Early Learning Then Overfitting p.56
- EM** Expectation Maximization - espérance maximisation p.41
- EMA** Exponential Moving Average) p.87
- EMV** Exponential Moving Variance p.87
- FBC** Frequency Band Correspondence - Correspondence en bande de fréquence p.109
- FBP** Filtered BackProjection - Rétroprojection filtrée p.36
- FDG** Fluorodésoxyglucose p.21
- FOV** Field Of View - Champ de vue p.26
- FWHM** Full Width at Half Maximum - Largeur à mi-hauteur p.65
- GT** Ground Truth - Vérité terrain p.56
- IR** Image Roughness - Rugosité d'image p.67
- KL** Divergence de Kullback-Leibler p.46
- LOR** Line Of Response - Ligne de réponse p.28
- MAP** Maximum A Posteriori p.40
- ML** Maximum Likelihood - Maximum de vraisemblance p.40
- MLE** Maximum Likelihood Estimation - Estimation du maximum de vraisemblance p.40
- MLEM** Maximum Likelihood Expectation Maximization p.41
- MRF** Markov Random Field - Champ aléatoire markovien p.48
- MSE** Mean Square Error p.55
- OSEM** Ordered Subsets Expectation Maximization p.42

- OSL** One Step Late p.45
- PML** Penalized Maximum Likelihood - Maximum de vraisemblance pénalisé p.41
- PMT** PhotoMultiplier Tube - Tube photomultiplicateur p.25
- PPGML** Penalized Preconditioned Gradient Maximum Likelihood p.45
- PSF** Point Spread Function - Fonction d'étalement du point p.33
- PSNR** Peak Signal to Noise Ratio p.87
- RDP** Relative Difference Penalty p.48
- ReLU** Rectified Linear Unit p.55
- ROI** Region Of Interest - Région d'intérêt p.66
- SC** Skip Connections - Sauts de connexion p.119
- SiPM** Silicon PhotoMultiplier - PhotoMultiplicateurs au Silicium p.25
- SSIM** mean Structural SIMilarity index p.66
- SSS** Single Scatter Simulation p.35
- SUV** Standardized Uptake Value - Valeurs de fixation normalisée p.64
- TEMP** Tomographie par Émission MonoPhotonique p.52
- TEP** Tomographie par Émission de Positons p.21
- TOF** Time Of Flight - Temps de vol p.29
- WMV** Window Moving Variance p.87

Par ailleurs, certains termes anglais ont été conservés volontairement dans ce manuscrit par manque de traduction satisfaisante en français. Certaines traductions ont été laissées dans les graphiques car ont été générés pour des conférences internationales. Ces termes sont listés ci-dessous :

- Max pooling,
- Inpainting,
- Deep Image Prior,
- Activity recovery : récupération d'activité,
- Image roughness : rugosité de l'image,
- Baseline : référence,
- Positive normalization : normalisation au maximum
- Mismatch : discordance
- Match : concordance

Introduction - Contexte - Objectifs de la thèse

La Tomographie par Emission de Positons (TEP) est une modalité d'imagerie fonctionnelle quantitative jouant un rôle important en oncologie en contribuant notamment au diagnostic et à la stadification des cancers ainsi qu'au suivi du traitement. Pour réaliser un examen TEP, un médicament radiopharmaceutique (aussi appelé radiotracteur) est d'abord injecté au patient. Ensuite, la caméra TEP détecte et compte les radiations issues des désintégrations radioactives du radiotracteur survenant dans le corps du patient. Ces données ainsi acquises, appelées données brutes, sont caractérisées par un bruit intrinsèque dû au principe de comptage et à la nature aléatoire des désintégrations radioactives. L'étape permettant d'obtenir une image de la distribution radioactive à partir des données acquises par la machine s'appelle la reconstruction tomographique. Le problème associé à la reconstruction tomographique est un problème mathématique appartenant à la catégorie des problèmes inverses mal posés. Cela signifie que pendant la résolution du problème, le bruit intrinsèque des données est amplifié et propagé dans l'image reconstruite. Il faut donc le contrôler pour que les images puissent être interprétables. La gestion du bruit et sa diminution se font au détriment de la quantification et entraînent ainsi du biais. Il existe différentes méthodes actuellement utilisées en clinique et chacune est caractérisée par un compromis entre biais et bruit dans les images reconstruites. Des algorithmes itératifs sont utilisés en routine clinique pour reconstruire une image TEP. Ceux-ci ont pour but de résoudre un problème d'optimisation mathématique dans le but de trouver une image correspondant au maximum de vraisemblance avec les données TEP, ou bien au maximum a posteriori, composé d'un terme d'attache aux données et d'un terme de régularisation. Dans le premier cas, un post traitement est utilisé à l'issue de la reconstruction pour lisser l'image afin de la rendre visuellement plus facile à interpréter. Dans le second cas, c'est le terme de régularisation qui limite le niveau de bruit dans l'image [71].

Dans cette thèse, nous avons voulu étudier le principe récemment proposé du Deep Image Prior (DIP) [92] utilisant un réseau de neurones de façon non supervisée. Cette méthode a été appliquée à une variété de tâches d'imagerie, et notamment au débruitage et à la reconstruction d'images TEP. La méthode DIPRecon [37] utilise un réseau DIP pour la reconstruction TEP et a permis d'obtenir des compromis entre biais et variance meilleurs que les algorithmes utilisés actuellement en clinique. Cependant, cet algorithme présente des biais intrinsèques dus à une contrainte de non-négativité sur l'image reconstruite. Dans ce contexte, nous proposons un nouvel algorithme de reconstruction appelé DNA, en tant qu'amélioration du DIPRecon, en autorisant les valeurs négatives dans l'image reconstruite. Pour supprimer la contrainte de non-négativité dans l'image, plusieurs algorithmes de la littérature ont été proposés, dont l'algorithme ADMM-Reg [59]. Ainsi, le DNA combine les algorithmes ADMM-Reg et DIPRecon en dérivant deux algorithmes ADMM (Alternating Direction Method of Multipliers) imbriqués [12].

Néanmoins, l'algorithme ADMM nécessite de régler judicieusement plusieurs hyperparamètres pour assurer sa convergence et pour qu'elle soit rapide. C'est aussi le cas de l'optimisation du réseau DIP. À notre connaissance, les méthodes actuelles de reconstruction utilisant un réseau DIP choisissent la plupart des paramètres manuellement, et ne s'accordent pas sur le choix de certains hyperparamètres. Le choix de tous ces hyperparamètres a une grande influence sur la qualité des images reconstruites. Certains hyperparamètres mal sélectionnés peuvent conduire à des images de faible qualité ou encore à une divergence de ce type d'algorithme. Dans ce travail,

nous avons ainsi réalisé une étude approfondie des hyperparamètres de l'algorithme DNA proposé. L'influence de chaque hyperparamètre a été étudiée pour être mieux comprise. Certains contribuent essentiellement à la vitesse de convergence de l'algorithme DNA, quand d'autres influent sur la capacité du réseau DIP à débruiter de façon qualitative et quantitative une image. Des intuitions pour en choisir judicieusement certains, ainsi que des méthodes inspirées de la littérature pour en régler d'autres de façon automatique ont été proposées.

Les scanners TEP sont quasiment toujours couplés en clinique à une modalité d'imagerie anatomique : tomodesitométrie (TDM) ou imagerie par résonance magnétique (IRM). Les images TEP sont caractérisées par une résolution spatiale de l'ordre de quelques millimètres alors que celle des images anatomiques est plutôt de l'ordre du millimètre. Ainsi, plusieurs propositions ont été faites dans la littérature pour incorporer des informations anatomiques dans la reconstruction TEP afin d'améliorer sa résolution spatiale et d'augmenter les détails structurels reconstruits. Le DIP permet notamment d'incorporer une information anatomique en entrée du débruitage. Plusieurs études ont montré que celle-ci apporte plus d'information de structures dans l'image TEP, ainsi que de meilleurs contrastes, notamment entre la matière blanche et la matière grise dans le cerveau en TEP FDG. Toutefois, l'utilisation d'information anatomique dans la reconstruction TEP pose un problème lorsqu'il y a des discordances entre information anatomique et image TEP. C'est notamment le cas lorsqu'une tumeur est seulement visible sur l'image anatomique. Toutes les méthodes proposées jusqu'alors peuvent générer des artefacts dans l'image TEP dans cette situation, ce qui compromet leur utilisation en clinique. Dans le cadre du DNA, nous avons proposé deux méthodes permettant d'éliminer ces artefacts tout en conservant un compromis entre biais et bruit meilleur que l'utilisation du DIP sans information anatomique.

Ce manuscrit s'articule en quatre chapitres :

- le premier traite des fondements de l'imagerie TEP, en passant par le principe physique des positons émis jusqu'à la détection d'événements stockés dans un format de données permettant de reconstruire une image. Plusieurs algorithmes de reconstruction d'intérêt, dont l'algorithme ADMM-Reg, sont détaillés dans ce chapitre,
- le deuxième introduit le principe du DIP ainsi que la méthode du DIPRecon, pour aboutir à la dérivation de l'algorithme DNA proposé dans cette thèse. Des simulations TEP 2D et des données réelles 3D ont été utilisées pour analyser le comportement qualitatif et quantitatif du DNA par rapport au DIPRecon et à d'autres algorithmes de reconstruction de l'état de l'art,
- le troisième étudie en détail chaque hyperparamètre du DNA à travers différentes expériences menées sur des données simulées. Il dresse un bilan sur la manière d'obtenir les meilleures performances de l'algorithme DNA,
- le quatrième propose deux méthodes intégrables à l'algorithme DNA permettant de bénéficier de l'information anatomique pour améliorer la qualité des images TEP reconstruite tout en empêchant la génération d'artefacts lors de la présence de discordances entre les deux modalités.

Chapitre 1

Imagerie TEP

1.1 Principe généraux TEP

La Tomographie par Émission de Positons (TEP) consiste à faire une image d'une distribution radioactive présente après l'injection d'un produit radioactif dans un patient. Le produit radioactif, aussi appelé radiopharmaceutique, se compose d'une molécule vectrice (désoxyglucose par exemple dans le cas du radiopharmaceutique le plus courant en imagerie TEP) associé à un isotope radioactif (^{18}F) dans le cas du Fluorodésoxyglucose (FDG). L'isotope se désintègre au cours du temps en émettant des positons. Ces positons s'annihilent avec des électrons de la matière environnante, en émettant deux rayons gamma en direction opposée. En les détectant avec la machine TEP, on peut déterminer une ligne sur laquelle a été annihilé le positon. Cela permet in fine de remonter à la position de l'isotope radioactif et du vecteur, et donc de reconstruire une image TEP de la distribution radioactive dans le patient.

Voyons dans le détail d'abord comment se désintègrent les isotopes radioactifs et comment les rayons gamma émis se déplacent dans le corps du patient.

1.1.1 Désintégration radioactive

1.1.1.1 Éléments physiques de l'isotope aux rayons gammas

Une désintégration β^+ est susceptible d'arriver lorsqu'un atome instable présente un excédent de protons. L'un de ces protons p peut se désintégrer et émettre 3 particules : un neutron n , un positon β^+ et un neutrino ν . On écrit cette désintégration comme ceci :



Les atomes de ce type sont ceux utilisés pour réaliser l'imagerie TEP (cf. tableau 1.1).

Isotope	Demi-vie ¹ (s)	Facteur d'embranchement ² pour la désintégration β^+
Cu-64	45720,	0,179
F-18	6586,2	0,967
Ga-68	4057,7	0,89
Rb-82	76,38	0,955
Zr-89	282276	0,227
Y-90	230590,8	0,00003186

TABLE 1.1 – Isotopes utilisés en TEP au CHU de Nantes et implémentés dans le logiciel de reconstruction CASToR (cf. Annexe C.1).

Le positon est l'anti-particule de l'électron : un positon a la même masse m_e qu'un électron, mais est de charge opposée. Le positon se thermalise puis disparaît par interaction avec un électron du milieu par annihilation. Celle-ci produit deux photons partant en directions opposées. Les deux photons ont une énergie égale à $E = m_e c^2 = 511\text{keV}$ chacun, avec c la célérité de la lumière dans le vide.

En pratique, les deux rayons gamma sont émis avec un angle légèrement différent de 180° . C'est ce qu'on appelle l'accolinéarité des rayons gamma émis. De plus, ceux-ci informent sur la position de l'annihilation positon-électron, et non pas exactement sur le lieu de la désintégration. La distance moyenne parcourue par le positon depuis le lieu de désintégration jusqu'au lieu de l'annihilation s'appelle le parcours du positon (positron range).

1.1.2 Interactions des photons gamma avec la matière

1.1.2.1 Effet photoélectrique

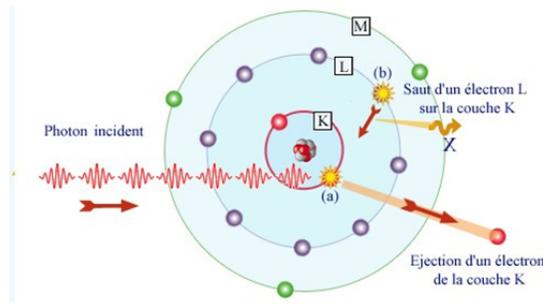


FIGURE 1.1 – Effet photoélectrique (source : www.laradioactivite.com).

L'effet photoélectrique est schématisé dans la figure 1.1 et expliqué ci-dessous. Un photon gamma rencontrant un atome peut interagir avec lui par effet photoélectrique et lui arracher un électron. Ce photon incident peut arracher des électrons des couches les plus internes de l'atome (couches K et L). Pour cela, toute l'énergie $E_{\text{photon}} = h\nu$ est absorbée par un électron de l'atome. Si cette énergie est supérieure à l'énergie de liaison E_l de l'électron, celui-ci a alors une énergie suffisante pour ne plus être lié à l'atome. Cet électron, appelé photoélectron, est alors arraché de l'atome avec une énergie $E_{\text{photoelectron}} = E_l - h\nu$. Un électron d'une couche supérieure descend alors sur une couche d'énergie plus faible pour stabiliser l'atome.

1.1.2.2 Effet Compton

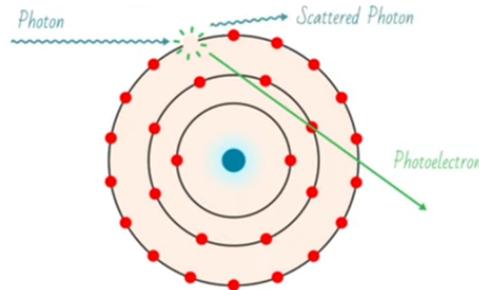


FIGURE 1.2 – Effet Compton (source : <https://www.youtube.com/watch?v=0j065KTznG8>).

1. Un atome radioactif se désintègre naturellement après un certain temps qui dépend de l'atome en question. Le temps au bout duquel la moitié des atomes radioactifs se sont désintégrés est appelé demi-vie.
2. La probabilité de se désintégrer sous un mode donné est appelé facteur d'embranchement.

L'effet Compton survient quand un photon interagit avec un électron de la couche externe d'un atome, comme sur la figure 1.2. Le photon cède une partie de son énergie initiale E_0 à cet électron qui est alors arraché de l'atome. On parle de diffusion Compton car en cédant son énergie, le photon est aussi dévié de sa trajectoire initiale avec un angle θ . On parle de photon diffusé (scattered photon). En utilisant les lois de conservation de la quantité de mouvement du photon et de l'énergie du système {photon, électron}, on peut déterminer l'énergie restante E du photon après interaction :

$$E = \frac{E_0}{1 + \frac{E_0}{m_e c^2} (1 - \cos \theta)} \quad (1.1)$$

avec m_e la masse de l'électron, c la célérité de la lumière, et le produit $m_e c^2 = 511$ keV.

Le photoélectron émis a donc perdu une énergie étant l'énergie initiale du photon moins ce qu'il lui reste, c'est-à-dire $E_0 - E$.

1.1.2.3 Diagramme de prédominance de ces effets

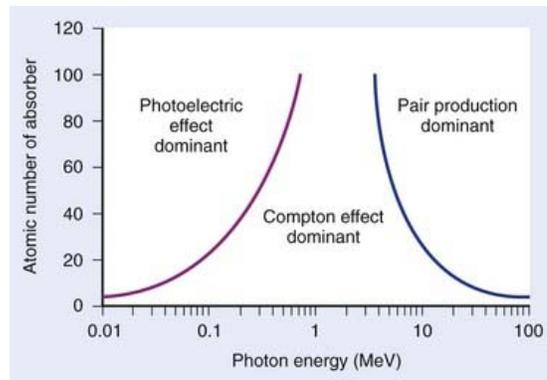


FIGURE 1.3 – Interactions de photons avec la matière en fonction de l'énergie du photon et du numéro atomique de la matière absorbante (source : <https://clinicalgate.com/principles-of-radiation-physics/>).

Ces différents types d'interactions de photons avec la matière sont plus ou moins probable en fonction de l'énergie du rayonnement ainsi que du numéro atomique de l'atome considéré. D'après la figure 1.3, à 511 keV, l'effet Compton est dominant dans des matériaux de faible numéro atomique, comme les tissus mous du corps humain ou dans l'eau. L'effet photoélectrique est donc très peu dominant dans le corps humain pendant un examen TEP. Il est dominant dans des matériaux de plus haut numéro atomique, comme les os ou le plomb. Il sera donc très probable dans les détecteurs de la machine TEP qui sont choisis avec des matériaux de numéro atomique élevé (cf. partie 1.1.3.1).

Remarques :

- la création de paire peut intervenir lorsqu'un photon traverse un atome, ce qui peut donner naissance à une paire électron-positon. Cela est seulement possible pour un photon de plus de 2×511 keV = 1.022 MeV,
- il existe aussi la diffusion de Rayleigh où un photon incident est simplement dévié de sa trajectoire en interagissant avec l'atome dans son ensemble, sans dépôt d'énergie. Cette interaction est importante pour des photons d'énergie inférieure à 50 keV. On ne s'intéresse donc pas à cet effet en TEP.

1.1.2.4 Modélisation de tous ces phénomènes : l'atténuation

Pour interpréter de façon macroscopique comment un ensemble de photons réagit à la traversée d'un matériau, on peut résumer l'interaction des photons à 3 cas : possibilité de traverser le matériau sans interaction, d'être absorbés par celui-ci, ou bien d'être diffusés. Ils sont pris en

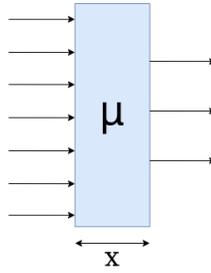


FIGURE 1.4 – Facteur d’atténuation μ pour modéliser le nombre de photons traversant une paroi d’épaisseur x sans interagir avec elle.

compte de façon globale par le phénomène d’atténuation (cf. figure 1.4). De façon mathématique, pour N_0 photons incidents à un milieu, on modélise le nombre de photons ayant traversé le matériau sans interaction par la relation suivante :

$$N = N_0 e^{-\mu x} \quad (1.2)$$

x est l’épaisseur du matériau, et $\mu = \tau + \sigma + \kappa$, avec μ le coefficient linéaire d’atténuation. Il prend en compte la contribution de l’effet photoélectrique, des éléments diffusés (Compton ou Rayleigh) et de la création de paires par l’intermédiaire de leurs coefficients d’atténuation propres, respectivement τ , σ et κ .

En résumé, l’isotope radioactif utilisé en TEP se désintègre par émission β^+ . Le positon alors émis se meut jusqu’à rencontrer un électron et s’annihiler, émettant deux rayons gamma de 511 keV qui subissent ou non des interactions avec le corps du patient. Pour reconstruire l’image, le scanner TEP doit ensuite détecter et compter ces photons émis pour être mis en relation entre eux et remonter à la position d’annihilation.

1.1.3 Fonctionnement d’un scanner TEP

Le système TEP possède des détecteurs dont le rôle est de détecter les photons issus des annihilations positon-électron et d’en compter le nombre. Pour pouvoir estimer l’origine du lieu d’annihilation, un circuit électronique est chargé d’associer deux photons venant du même lieu d’annihilation.

Dans la partie suivante, nous présenterons comment fonctionnent les machines TEP du CHU de Nantes à l’heure actuelle. Leurs détecteurs sont composés d’un cristal scintillateur et d’un photodétecteur de type PMT ou SiPM que nous allons détailler par la suite.

1.1.3.1 Cristaux scintillateur

Les détecteurs TEP sont faits de cristaux scintillateurs, permettant d’émettre un grand nombre de photons proche-visibles suite à l’absorption des photons de plus haute énergie issus de l’annihilation positon-électron. En effet, les photons incidents interagissent avec le matériau scintillateur par effet Compton ou photoélectrique, ce qui résulte en l’émission d’électrons. Les molécules du matériau scintillateur passent alors à un état excité, avec donc des niveaux d’énergie plus élevés. La désexcitation de ces molécules vers leur état fondamental émet de l’énergie sous la forme de photons de moindre énergie, qui sont dans le spectre du proche-visible pour un matériau scintillateur.

Les matériaux scintillateurs les plus populaires aujourd’hui sont reportés dans le tableau 1.2 :

Le choix du matériau scintillateur pour les détecteurs est fait selon plusieurs critères à privilégier en fonction de l’application [76] :

- *résolution énergétique*. Chaque matériau scintillateur a une résolution énergétique intrinsèque de part ses propriétés physiques. Elle doit être la plus petite possible,

Matériau	Formule chimique développée	Rendement lumineux (nombre de photons par keV)	Constante de temps de scintillation	Numéro atomique effectif
BGO	Bi ₄ Ge ₃ O ₁₂	6	300 ns	74
LSO	Lu ₂ SiO ₅ : Ce	29	40 ns	66
LYSO	LuYSiO ₅ : Ce	25	50 ns	65
GSO	Gd ₂ SiO ₅ : Zr	10	50 ns	59

TABLE 1.2 – Différents matériaux scintillateurs utilisés en TEP [76]. Abréviations : BGO = germanate de bismuth, LSO = orthosilicate de lutétium, LYSO = Orthosilicate de lutétium et d’yttrium, dopé au Cerium, GSO = orthosilicate de gadolinium, dopé au Cerium.

- *rendement lumineux*. Le nombre de photons visibles émis après l’absorption d’un rayon gamma dépend du matériau scintillateur. Ce nombre doit être le plus grand possible pour un bon rendement lumineux,
- *résolution temporelle*. Chaque matériau scintillateur possède un temps en moyenne plus ou moins long pour scintiller. Il doit être le plus court possible pour une meilleure résolution temporelle,
- *pouvoir stoppant du détecteur*. Il correspond à la distance moyenne que parcourt un photon dans le matériau scintillateur avant d’être absorbé par celui-ci. Cette distance doit être la plus courte possible pour un fort pouvoir stoppant, ce qui est favorisé par un matériau de haute densité et de grand numéro atomique effectif,

1.1.3.2 Photodétecteurs

Le but d’un photodétecteur est de créer un signal électrique en détectant et transformant un photon visible en électrons.

1.1.3.2.1 Tube Photomultiplicateur (PMT) Le tube photomultiplicateur (PMT : PhotoMultiplier Tube) utilise l’effet photoélectrique. Le photon en sortie du cristal scintillateur rencontre le premier élément du PMT qui est une photocathode, ce qui produit un photoélectron. Celui-ci continue ensuite son chemin dans le PMT et subit des champs électriques pour être dévié de sa trajectoire initiale et rencontrer des dynodes. Le photoélectron incident est alors accéléré par un premier champ électrique au niveau de la première dynode (d’une tension autour de 100V), puis perd toute son énergie pour arracher plusieurs électrons de cette dynode. Ceux-ci sont donc réémis avec une moindre énergie, puis à nouveau accélérés par un deuxième champ électrique au niveau de la deuxième dynode (avec un incrément de 100V) arrachant encore plus d’électrons. Une fois toutes les dynodes atteintes, de nombreux électrons se présentent au niveau d’une anode à l’arrivée du PMT. Cette accumulation de charge crée une tension au niveau de l’anode, qui est transmise à un préamplificateur puis à un amplificateur pour avoir une impulsion électrique suffisante. Celle-ci est enfin transmise au Pulse Height Analyzer (PHA) qui se charge d’analyser l’intensité de l’impulsion électrique pour déterminer l’énergie initiale du rayon gamma incident. De cette façon, le système peut discriminer les rayons gamma incidents.

1.1.3.2.2 PhotoMultiplicateurs au Silicium (SiPM) Plus récemment, des PhotoMultiplicateurs au Silicium (SiPM Silicon PhotoMultiplier) ont été développés. Ceux-ci sont formés d’une matrice de photodiodes à avalanches fonctionnant en mode Geiger-Müller.

Les SiPM nécessitent une tension d’alimentation plus faible que les PMT, et sont très peu sensibles à la présence d’un champ magnétique. Cette dernière propriété a pour avantage de pouvoir coupler le détecteur TEP à un IRM présentant un champ magnétique élevé, sans que la détection TEP soit perturbée, contrairement aux PMT. Ils présentent aussi comme avantage d’avoir une résolution temporelle en dessous de la nanoseconde [8], ce qui est bien inférieur à celle des PMT (cf. tableau 1.2).

1.1.3.3 Organisation d'un scanner TEP

1.1.3.3.1 Organisation en blocs détecteurs Dans les premiers scanners TEP, un PMT était associé à un cristal (one-to-one coupling en anglais). Le multiplexing a été ensuite utilisé, en associant plusieurs cristaux scintillateurs à un même PMT. On appelle bloc détecteur l'ensemble de plusieurs PMT juxtaposés, chacun associé à plusieurs cristaux (cf. figure 1.5). Les PMT reçoivent donc une information lumineuse pouvant provenir de plusieurs cristaux différents (phénomène de light sharing). L'utilisation de blocs détecteurs diminue les coûts de la machine car moins de PMT sont utilisés. Le nombre de signaux électriques à traiter en parallèle est aussi diminué car moins de PMT et de chaînes électroniques pour traiter le signal électrique en sortie du PMT. En revanche, pour un scanner TEP utilisant des blocs détecteurs, l'énergie d'un photon incident peut être absorbée par plusieurs cristaux se partageant un même PMT. La position d'interaction est estimée par le barycentre des cristaux pondérés par les énergies absorbées par chaque cristal. L'utilisation de blocs détecteurs engendre une perte de résolution spatiale liée à l'estimation de la position d'interaction du photon [91].

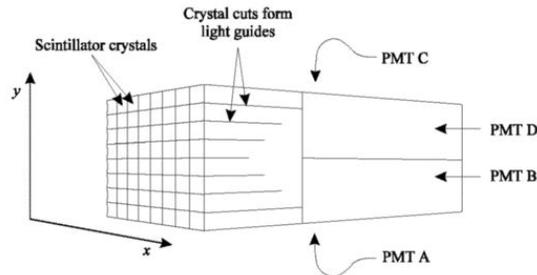


FIGURE 1.5 – Organisation d'un bloc détecteur. La figure est extraite de l'article [11].

1.1.3.3.2 Organisation en anneau Pendant un examen TEP, le patient est allongé sur un lit à l'intérieur de la machine TEP. La direction le long du corps du patient de haut en bas est appelée la direction axiale. L'axe axial est noté l'axe z et est perpendiculaire au plan transverse en vert, comme montré sur la figure 1.6.

Les scanners TEP usuels (tel que ceux installés au CHU de Nantes) sont généralement constitués d'anneaux de détection afin de pouvoir détecter les photons gammas arrivant depuis n'importe quelle direction à l'intérieur du scanner. Un anneau contient des blocs détecteurs juxtaposés tout autour du patient selon un plan transverse comme montré sur la partie gauche de la figure 1.7. Plusieurs anneaux sont agencés dans plusieurs plans transverses (cf. figure 1.7) définissant ainsi le champ de vue axial (Field Of View - FOV) pouvant être imagé.

Une annihilation survenant trop loin du centre du scanner TEP ne peut pas provenir du patient. En effet, le volume occupé par le patient est inférieur au volume disponible à l'intérieur

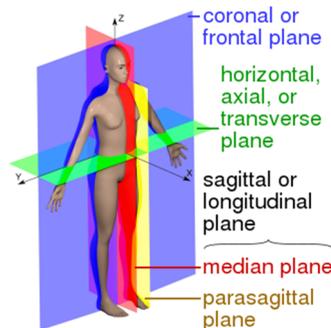


FIGURE 1.6 – Différents axes et plans à travers le corps humain. En particulier, la direction axiale est perpendiculaire au plan transverse (source : www.wikipedia.org).

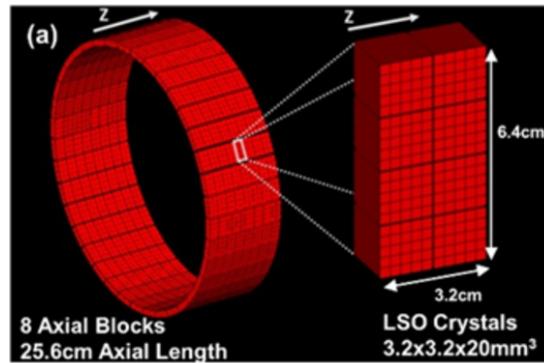


FIGURE 1.7 – Blocs détecteurs et anneaux du scanner TEP Siemens Biograph Vision. La partie de gauche de la figure montre l’agencement des blocs détecteurs en anneau, ainsi que plusieurs anneaux de détection juxtaposés selon la direction axiale z . La partie droite de cette figure montre un bloc détecteur formés de plusieurs matrices de cristaux (source : <https://radiology.weill.cornell.edu/research/research-projects/long-axial-field-view-pet>).

du scanner TEP. Ainsi, le champ de vue du scanner TEP est délimité par l’espace 3D formé par l’ensemble des blocs détecteurs, auxquelles on retire l’espace reliant deux blocs détecteurs trop proches selon un plan transverse.

1.1.4 Évènements enregistrés par la machine TEP

1.1.4.1 Détections simples

Chaque photon arrivant au niveau d’un détecteur de la machine TEP et effectivement détecté par celui-ci donne lieu à une détection simple. En pratique, de nombreux rayons gamma arrivent au niveau des détecteurs mais n’ont pas forcément une énergie exactement de 511 keV. De plus, le détecteur possède en pratique une résolution énergétique limitée à cause des propriétés intrinsèques du cristal scintillateur utilisé dans le détecteur (cf. partie 1.1.3.1). Pour cela, une fenêtre en énergie est utilisée pour accepter ou rejeter une détection simple.

1.1.4.2 Coïncidences

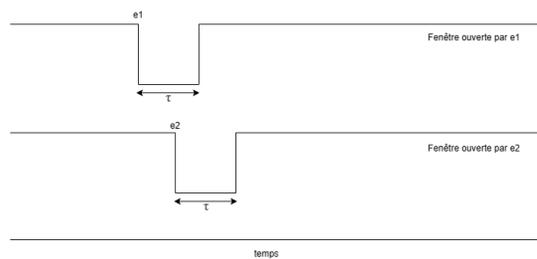


FIGURE 1.8 – Fenêtre temporelle ouverte à chaque détection simple, ici e_1 et e_2 .

Les données TEP qui seront utilisées pour la reconstruction correspondent à l’association de deux détections simples ensemble pour former ce qu’on appelle une coïncidence. Pour cela, à chaque évènement e détecté par la machine TEP, une fenêtre temporelle de durée τ est ouverte (cf. figure 1.8). Cette fenêtre est appelée fenêtre de coïncidence. Si un autre évènement ouvre une fenêtre en même temps, alors les deux évènements sont mis en coïncidence par le système. La durée τ est choisie pour que deux rayons gamma issus d’une même annihilation aient le temps d’être détectés par la machine TEP (le pire des cas étant le temps que parcourrait un rayon gamma le long d’un diamètre oblique allant du premier au dernier anneau du scanner TEP). Si

deux rayons gamma arrivent avec un écart de plus de τ , alors ils ne peuvent pas provenir de la même annihilation et ne sont donc pas mis en coïncidence.

L'ensemble de toutes les coïncidences détectées se nomme les prompts.

1.1.4.2.1 Coïncidences vraies Une coïncidence est dite vraie (true coincidence) si la machine TEP associe deux rayons gamma provenant bien d'une même annihilation, sans avoir été diffusé dans le corps du patient. On sait alors que la position de l'annihilation est dans le volume reliant les deux détecteurs, ce qu'on assimile en pratique à une ligne, appelée ligne de réponse (Line Of Response - LOR). LOR

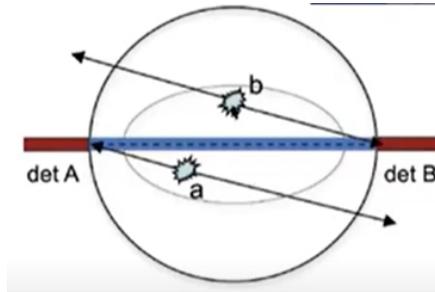


FIGURE 1.9 – Coïncidence fortuite : deux détections simples venant de deux annihilations différentes sont mises en coïncidence. La figure est extraite de l'article [20].

1.1.4.2.2 Coïncidences fortuites Une coïncidence est dite fortuite (random coincidence en anglais) lorsque deux détections simples sont mises en coïncidences, alors que les deux photons proviennent de deux annihilations différentes (cf. figure 1.9). Deux annihilations de ce type peuvent arriver dans un intervalle de temps inférieur à le fenêtre de coïncidence τ , et alors donner lieu à une coïncidence sur une LOR fictive.

En particulier, un cas de détections fortuites peut arriver en utilisant des détecteurs avec du Lutétium (*Lu*) comme les détecteurs LSO ou LYSO. En effet, ces détecteurs utilisent du *Lu* extrait naturellement. Cependant, sur Terre, il existe 2.6% de ^{176}Lu qui est un isotope radioactif, avec notamment une émission β^- , puis désexcitation gamma de 307 et 202 keV possibles. La somme de deux rayons gamma de ce type est de 509 keV et tombe donc dans le fenêtre d'énergie [3].

La proportion de coïncidences fortuites sur les prompts est appelée la fraction de fortuites. Le nombre de coïncidences vraies auxquelles ont été retirées le nombre de coïncidences fortuites est appelée le nombre de net trues.

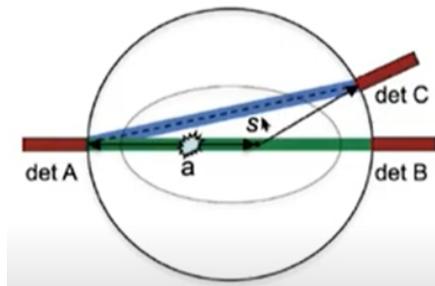


FIGURE 1.10 – Coïncidence diffusée : l'un ou les deux photons ont été diffusés avant d'être détectés. La figure est extraite de l'article [20].

1.1.4.2.3 Coïncidences diffusées Une coïncidence diffusée (scatter coincidence) est formée à partir de deux détections simples venant d'une même annihilation, dont l'un ou les deux pho-

tons ont été diffusés dans le corps du patient par effet Compton. Dans ce cas, la machine TEP forme une LOR entre deux détecteurs dont l'un ou les deux n'auraient pas dû détecter de photons s'il n'y avait pas eu de diffusion (cf. figure 1.10).

Une coïncidence diffusée ne peut pas être fortuite car les deux photons détectés proviennent de la même annihilation. On définit alors la fraction de diffusées comme la proportion de coïncidences diffusées sur la proportion de net trues.

1.1.4.2.4 Coïncidences multiples Enfin, une coïncidence est dite multiple si au moins 3 détections simples ont lieu dans la même fenêtre de coïncidence.

1.1.4.3 Temps de vol (TOF)

A partir d'une coïncidence vraie, on peut savoir que le lieu d'une annihilation est sur une LOR formée par deux détecteurs de la machine TEP. Pour gagner en précision sur la position de lieu d'annihilation, les scanners TEP les plus récents peuvent aussi mesurer l'intervalle de temps séparant deux détections simples formant une coïncidence. L'intervalle de temps les séparant renseigne sur la distance parcourue par les deux rayons gamma et donc sur le lieu d'annihilation. Cette information supplémentaire s'appelle le temps de vol (Time Of Flight - TOF). En revanche, en pratique, la résolution temporelle du TOF est limitée, ce qui améliore la connaissance du lieu d'annihilation mais sans être parfaite. Les scanners commerciaux actuels ont une résolution temporelle allant de 600 ps à 200 ps (550 ps pour le Siemens Biograph mCT, 214 ps pour le Siemens Biograph Vision). [77]

Ainsi, le scanner TEP détecte les photons provenant du corps du patient et forme des coïncidences pour associer les rayons gamma entre eux. Voyons maintenant de quelle façon sont stockées les coïncidences.

1.1.5 Principe d'acquisition

1.1.5.1 Format sinogramme

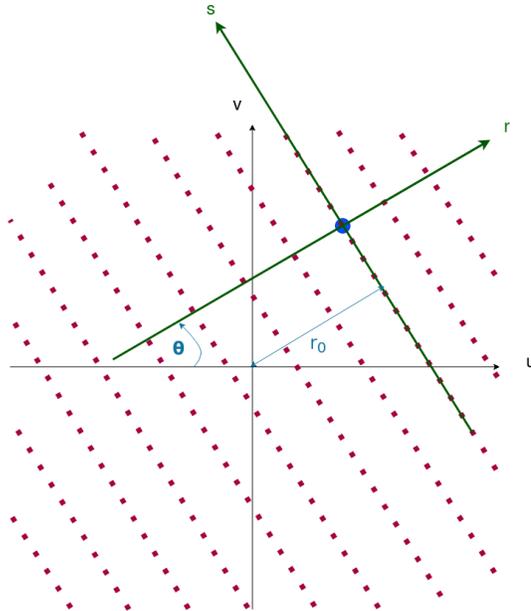


FIGURE 1.11 – Transformée de Radon $R(x(r, \phi))$ pour un point de radioactivité (en bleu) dans l'espace de coordonnées (u, v) .

L'enregistrement des données TEP en format sinogramme a été imaginé en s'appuyant sur la transformée de Radon. Un élément (ou bin) du sinogramme correspond au nombre de coïncidences enregistrées sur une LOR, elle-même définie par un angle ϕ et une distance au centre du scanner r (cf. figure 1.11).

Remarque : dans cette partie, nous détaillerons comment sont enregistrées les sinogrammes 2D, permettant de reconstruire des images 2D.

1.1.5.1.1 Transformée de Radon Considérons l'intégrale de la distribution radioactive x sur une LOR L :

$$R_x(L) = \int_L x(u, v) \sqrt{(du^2 + dv^2)}$$

Pour transformer cette intégrale, on peut réaliser une transformation passive du vecteur $\begin{pmatrix} u \\ v \end{pmatrix}$ en passant de ses coordonnées cartésiennes à ses coordonnées polaires. Cette transformation est dite "passive" car elle consiste à effectuer une rotation des axes cartésiens aux axes polaires, sans modifier la position du vecteur (contrairement à une transformation active où le vecteur tourne dans une base fixe, ce qui change ces coordonnées). Les nouvelles coordonnées sont alors données par la rotation passive :

$$\begin{pmatrix} r \\ s \end{pmatrix} = Q^T \begin{pmatrix} u \\ v \end{pmatrix}$$

avec $Q = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix}$ la matrice de rotation 2D d'angle ϕ .

Ainsi, on définit la transformée de Radon par l'intégrale suivante :

$$R(x(r_0, \phi)) = \int_{-\infty}^{+\infty} x(r_0 \cos \phi - s \sin \phi, r_0 \sin \phi + s \cos \phi) ds$$

Dans le cas de la TEP, s est la variable de déplacement le long d'une LOR, ϕ l'angle de la LOR avec l'axe des abscisses, et r_0 la distance minimale entre la LOR et le centre du scanner (cf. figure 1.11). r_0 et ϕ sont donc fixés pour une LOR, seul s varie le long de la LOR.

1.1.5.1.2 Sinogramme 2D Pour comprendre la construction d'un sinogramme 2D, considérons pour le moment que seules des coïncidences vraies sont formées lors d'une acquisition TEP. Négligeons aussi le phénomène d'atténuation et supposons que l'efficacité de détection est idéale.

Un sinogramme 2D est une matrice 2D dont les lignes stockent toutes les transformées de Radon pour chaque valeur de r_0 et pour un angle ϕ fixé. Les colonnes stockent les transformées de Radon pour tous les angles ϕ et pour une distance au centre r_0 fixe.

La transformée de Radon en TEP étant l'intégrale de la distribution radioactive le long d'une LOR, elle correspond donc à la somme de toutes les annihilations arrivant entre les deux détecteurs formant la LOR.

Les annihilations le long d'une LOR sont identifiées par le système TEP par les coïncidences détectées le long de cette LOR. Ainsi, un sinogramme stocke le nombre de coïncidences vraies pour chaque LOR, ce qui correspond à stocker les transformées de Radon le long d'une LOR. En pratique, on ne stocke qu'un ensemble d'angles ϕ et de distances au centre r_0 fini, définis par la disposition et le nombre de détecteurs du scanner TEP.

Par exemple, le sinogramme 2D correspondant au point de radioactivité de la figure 1.11 est donné par la figure 1.12.

Cette description des sinogrammes à partir de la transformée de Radon ne représente pas fidèlement la réalité car nous avons émis des hypothèses préliminaires en début de partie. Cela nous a permis d'expliquer l'intuition derrière le format de sinogramme. En réalité, on stocke plusieurs données différentes sous ce format : le sinogramme des prompts, le sinogramme

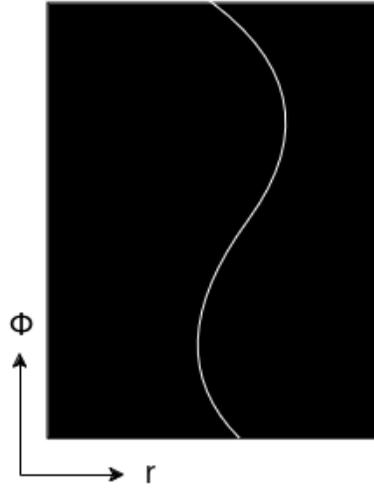


FIGURE 1.12 – Sinogramme 2D pour un point de radioactivité. Seules les valeurs le long d’une ligne sinusoïdale blanche sont non nulles.

d’atténuation, le sinogramme de normalisation, le sinogramme des coïncidences fortuites, et le sinogramme des coïncidences diffusées. Nous allons voir par la suite comment ils sont obtenus (cf. partie 1.2.1.2).

1.1.5.2 Format mode liste

Un sinogramme étant une accumulation des coïncidences sur un temps donné, il peut contenir des éléments vides si le nombre de coïncidences est nul sur une LOR pendant l’acquisition. Pour un scanner TEP, le nombre de coïncidences dépend de l’activité dans le champ de vue, du facteur d’embranchement du radioisotope, et de la durée de l’acquisition. Ce cas de figure peut arriver dans le cas de faibles statistiques d’acquisition, c’est-à-dire une acquisition avec peu de coïncidences vraies.

Plutôt que d’être agencées géométriquement comme dans un sinogramme, les données peuvent être enregistrées en mode liste. Chaque coïncidence est enregistrée par ordre d’arrivée et avec des données supplémentaires nécessaires pour pouvoir reconstruire l’image (indices des détecteurs, temps d’arrivée et énergies des rayons gamma incidents, temps de vol). Le format de stockage mode liste peut donc permettre d’économiser de la place en mémoire.

Dans cette thèse, nous nous concentrerons seulement sur l’acquisition et la reconstruction avec données en format sinogramme car l’algorithme développé (cf. partie 2 ainsi que plusieurs algorithmes utilisés pour les comparaisons ne sont utilisables qu’avec des données en format sinogramme.

1.1.5.3 Données temps de vol

Les données TOF peuvent être traitées aussi bien à partir du format mode liste que du format sinogramme. L’incertitude sur la position de l’annihilation est modélisée non plus de façon uniforme sur toute la LOR comme pour des données sans TOF, mais par une distribution de probabilité appelée noyau TOF (cf. Figure 1.13). Pour le format sinogramme, une LOR est découpée en plusieurs éléments appelés TOF bins [33]. On parle de sinogramme 4D pour un scanner TEP 3D utilisant des données TOF, car une dimension supplémentaire est ajoutée au sinogramme 3D.

Une fois les données acquises par la machine TEP, il faut établir un modèle mathématique permettant de remonter à la distribution radioactive dont on souhaite obtenir une image. Tout

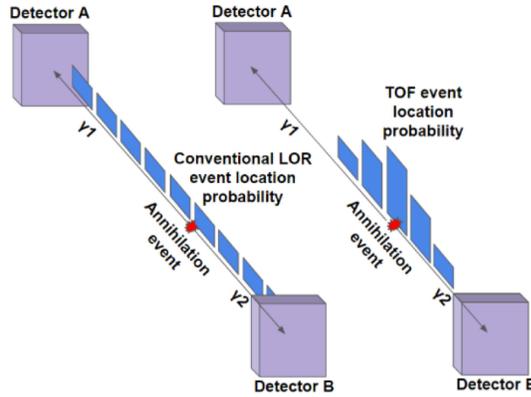


FIGURE 1.13 – Probabilité d’une annihilation sur une LOR sans information TOF par rapport à une LOR avec information TOF. La figure est extraite de l’article [57].

d’abord, nous allons voir le modèle mathématique utilisé, ainsi que les corrections à effectuer pour bien modéliser l’acquisition TEP.

1.2 Des données acquises à l’image reconstruite

1.2.1 Modèle direct et corrections

Dans cette partie, nous allons introduire mathématiquement le lien entre les données acquises par la machine, notées avec le vecteur y , et l’image de la distribution radioactive dans le patient, notée par le vecteur x .

Remarque : la notation vecteur sera utilisée dans tout le manuscrit comme fait habituellement dans le domaine de la TEP, bien que les sinogrammes y et les images x soient visualisées en 2D ou 3D.

1.2.1.1 Modèle direct

Pendant l’acquisition TEP, chaque bin de sinogramme i compte le nombre de coïncidences y_i dans la LOR i . Soit Y_i la variable aléatoire comptant le nombre de coïncidences dans la LOR i , de moyenne \bar{y}_i . On cherche à modéliser le nombre de coïncidences \bar{y}_i détectées en moyenne sur la LOR i . Pour cela, il faut calculer la concentration radioactive présente sur la LOR i . Il faut aussi modéliser les effets physiques de déplacement de ces photons, ainsi que la réponse des détecteurs à l’arrivée de ces photons. Le modèle direct modélise cela grâce à l’équation suivante [71] le long de la LOR i :

$$\bar{y}_i = \left(\int_{\text{FOV}} x^{(c)}(l) c(i, l) dl \right) + \bar{b}_i$$

avec $x^{(c)}$ la distribution radioactive, l représentant la variable de déplacement sur tout le champ de vue (FOV), c la fonction de réponse des détecteurs sur la LOR i en fonction aussi de la position dans l’image de la désintégration, et \bar{b}_i la contribution moyenne des coïncidences fortuites et diffusées.

Une distribution radioactive $x^{(c)}$ est un objet dans un espace de dimension infinie. En pratique, le modèle continu est discrétisé par le choix d’un nombre fini de fonctions de base $\Phi_j^{(c)}$ (dans ce travail, des fonctions portes qui forment donc des voxels - l’équivalent des pixels en 3D -). La distribution radioactive discrétisée est notée par le vecteur $x = (x_j)_j$ et peut se décomposer de façon approchée sur cette base selon la formule suivante :

$$x^{(c)} \approx \sum_j x_j \Phi_j^{(c)} \quad (1.3)$$

Cela permet de réécrire le modèle direct utilisé en pratique de façon discrète :

$$\bar{y}_i = \left(\int_{\text{FOV}} x^{(c)}(l) c(i, l) dl \right) + \bar{b}_i \quad (1.4)$$

$$= \left(\int_{\text{FOV}} \left(\sum_j x_j \Phi_j^{(c)}(l) \right) c(i, l) dl \right) + \bar{b}_i \quad (1.5)$$

$$= \sum_j \left(\int_{\text{FOV}} \Phi_j^{(c)}(l) c(i, l) dl \right) x_j + \bar{b}_i \quad (1.6)$$

$$= \sum_j A_{ij} x_j + \bar{b}_i \quad (1.7)$$

avec $A_{ij} = \int_{\text{FOV}} \Phi_j^{(c)}(l) c(i, l) dl$ un terme proportionnel à la probabilité qu'une désintégration arrivant dans le voxel j soit détectée par la machine TEP sur la LOR i .

Par la suite, nous allons utiliser le modèle direct (1.7) qui s'écrit aussi de façon vectorielle :

$$\bar{y} = Ax + \bar{b} \quad (1.8)$$

Ainsi, après discrétisation, on obtient le nombre de coïncidences pour chaque LOR (en moyenne) à partir de la distribution radioactive x dans le patient à laquelle on applique une transformation affine. Le coefficient multiplicateur A est appelé matrice système, et \bar{b} représente le nombre de coïncidences fortuites et diffusées (en moyenne). Nous allons voir dans les parties suivantes comment déterminer ces termes et finalement obtenir l'image x .

1.2.1.2 Termes de correction

La matrice système modélise le lien entre les données TEP et l'image de façon multiplicative, contrairement au terme de coïncidences fortuites et diffusées qui est modélisé sous forme additive dans le modèle direct. Ces termes sont estimés préalablement à la reconstruction en utilisant la géométrie du scanner TEP et la physique des particules.

1.2.1.2.1 Matrice système D'après l'écriture du modèle direct, chaque élément A_{ij} de la matrice système est proportionnel à la probabilité qu'une annihilation survenue dans le voxel x_j soit détectée par la machine TEP sur la LOR i . Elle permet de modéliser de façon multiplicative certains phénomènes contribuant à la détection d'une annihilation.

Modélisant plusieurs phénomènes agissant de façon multiplicative qui sont indépendants, la matrice système A peut se décomposer selon plusieurs matrices [73] :

$$A = P^{(s)} N L X P^{(i)} \quad (1.9)$$

avec $P^{(s)}$ et $P^{(i)}$ les termes contribuant à la Point Spread Function (PSF) respectivement dans l'espace sinogramme et image, N au terme de normalisation et X au terme géométrique. Détaillons chacun de ces termes dans les parties suivantes.

1.2.1.2.1.1 Point Spread Function Pour une image x qui serait en réalité un point, le système TEP la voit comme une tâche (cf. figure 1.14). Pour modéliser ce phénomène de façon globale, on définit la fonction d'étalement du point notée PSF (Point Spread Function en anglais) du système, qui prend en compte tous les effets diminuant la résolution spatiale de l'image finale sous la forme d'une convolution. Plusieurs phénomènes physiques y contribuent : parcours du positon, accolinéarité des photons émis, diffusion inter-cristaux, pénétration dans un cristal, taille des détecteurs, décodage électronique, erreur d'échantillonnage [63].

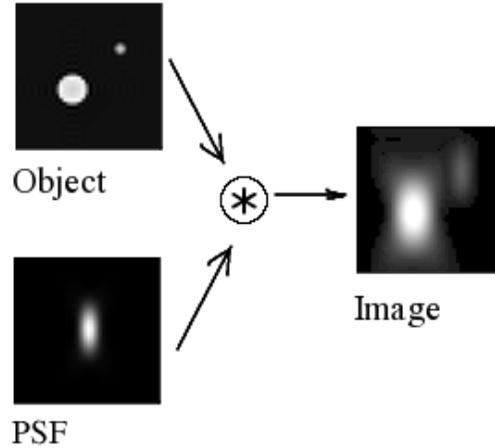


FIGURE 1.14 – Deux objets circulaires vue de façon floue par le système TEP, modélisé par la PSF (source : www.wikipedia.org).

1.2.1.2.1.2 Projecteur géométrique Le changement de coordonnées entre image et sinogramme s’effectue par la transformée de Radon (cf. partie 1.1.5.1.1). En pratique, la transformée de Radon est discrétisée et ce changement de coordonnées est effectué par un algorithme appelé projecteur géométrique. Un exemple de projecteur a été proposé dans [84] [54].

Remarque : dans la suite du manuscrit, nous parlerons de projection pour parler plus simplement du produit matriciel de la matrice système A par un vecteur dans l’espace image (la distribution radioactive x par exemple). Le résultat est dans l’espace des projections, qui peut être vu comme un sinogramme. L’opération de rétroprojection est le produit matriciel de la transposée de la matrice système (notée A^T) par un vecteur dans l’espace sinogramme (les données acquises y par exemple). Le résultat est dans l’espace image.

1.2.1.2.1.3 Correction d’atténuation Étant donné la distribution de radioactivité dans le corps x , un certain nombre de coïncidences devraient être détectées dans la LOR i . Mais en réalité, le système TEP voit seulement les coïncidences où les 2 photons issus de la même désintégration ne sont pas diffusés ou absorbés. Pour prendre en compte tous les cas où au moins un photon est diffusé ou absorbé et n’est pas vu par le système TEP, on inclut un terme multiplicatif dans la matrice système qui sert à effectuer une correction d’atténuation (diffusion ou absorption, cf. partie 1.1.2.4).

On associe donc à chaque LOR un facteur de correction d’atténuation (ACF : Attenuation Correction Factor). Pour le calculer, il suffit de remarquer qu’il ne dépend pas du point d’annihilation car la longueur traversée par les deux rayons gamma est toujours la distance entre les deux détecteurs formant la LOR. Il ne dépend donc que des coefficients linéaires d’atténuation des matériaux du corps sur cette LOR.

La carte d’atténuation, ou μ -map, correspond à une image des coefficients linéaires d’atténuation dans le champ de vue du scanner TEP. Elle peut être calculée à partir de l’image TDM pour un examen TEP/TDM, ou en utilisant des méthodes spécifiques s’appuyant sur l’image IRM pour un examen TEP/IRM [21]. Pour calculer l’ACF pour chaque LOR, il faut préalablement le noyau de convolution de la PSF à la μ -map pour mettre l’image anatomique à la résolution spatiale de la TEP. Ensuite, il suffit d’en prendre la projection géométrique, ce qui permet de déterminer (de façon approchée dû à la discrétisation de la matrice système) l’intégrale des coefficients d’atténuation sur cette même LOR. Ainsi on obtient un sinogramme d’ACF à partir de la μ -map :

$$ACF = \exp(-A\mu_{\text{map}}) \quad (1.10)$$

1.2.1.2.1.4 Normalisation La sensibilité correspond à la capacité d'un scanner TEP à détecter le plus possibles de coïncidences émises. Les détecteurs d'un scanner TEP sont caractérisés en pratique par une sensibilité variable dépendant de plusieurs facteurs. Premièrement, des facteurs géométriques contribuent à la sensibilité du scanner TEP. En effet, la sensibilité dépend de la position des détecteurs et de leur distance par rapport au centre du champ de vue. La sensibilité est plus grande si la ligne entre 2 détecteurs d'une LOR est dans la même direction que l'inclinaison des détecteurs. En effet, dans ce cas, les rayons gamma ont plus de distance pour être absorbés par le détecteur. Les détecteurs ont aussi une efficacité de détection intrinsèquement différente qu'il faut prendre en compte. De plus, les fenêtres de coïncidences de tous les détecteurs doivent être parfaitement synchronisées pour l'attribution d'une coïncidence à la bonne LOR [4]. Enfin, un certain temps s'écoule entre l'arrivée d'un rayon gamma et la fin de son traitement par le système. Ce temps est appelé temps mort, car le système ne peut plus voir d'autres éventuelles arrivées de rayons gamma survenant pendant ce laps de temps. Ces temps morts affectent la sensibilité d'un scanner TEP. Un sinogramme de normalisation doit alors être estimé pour normaliser ces différences sur chaque LOR. Pour corriger tous ces effets, il existe par exemple les méthodes de type component-based normalization, qui séparent le terme de normalisation de la matrice système en plusieurs facteurs [5].

1.2.1.2.2 Correction des coïncidences diffusées L'estimation du sinogramme des coïncidences diffusées est effectuée sur les scanners actuels soit par :

- des simulations analytiques de la physique à partir de la formule de Klein-Nishina,
- des méthodes Monte Carlo simulant l'émission de particules une par une et leur trajectoire dans le patient,
- ou des méthodes hybrides telle que la méthode SSS (Single Scatter Simulation) [103] utilisée par les machines Siemens au CHU de Nantes.

Le sinogramme des coïncidences diffusées obtenu contribue au terme \bar{b} de l'équation (1.8).

1.2.1.2.3 Correction des coïncidences fortuites Il existe deux méthodes principales pour obtenir un sinogramme estimé des coïncidences fortuites.

La première estime directement le nombre moyen de coïncidences fortuites entre deux détecteurs à partir de la mesure du taux de comptage de détections simples de chaque détecteur. La formule théorique est démontrée dans le cas général page 792-3 dans [31], ce qui donne pour le cas de la TEP [86] :

$$\bar{r}_{ab} = 2\tau s_a s_b$$

où \bar{r}_{ab} est l'estimation du nombre moyen de coïncidences fortuites entre les détecteurs a et b , τ est la durée de la fenêtre de coïncidence, et s_a , s_b sont les taux moyens de détections simples respectivement dans les détecteurs a et b . Cependant, le temps mort pour traiter une coïncidence peut être plus long qu'un temps mort pour traiter une détection simple [70], biaisant alors l'estimation.

La deuxième méthode est appelée fenêtre de retard. Après une détection simple, une fenêtre de coïncidence d'une durée τ est ouverte (cf. partie 1.1.4.2). Une fenêtre, appelée fenêtre de retard, de même durée τ que la fenêtre de coïncidence est aussi ouverte après un temps $\tau_r > \tau$ assez grand pour rendre impossible la détection d'une coïncidence vraie ou diffusée. Ces coïncidences formées par la fenêtre de retard sont par définition des coïncidences fortuites : deux photons venant de deux annihilations différentes. Par cette méthode, on obtient une estimation non biaisée du nombre de coïncidences fortuites. Cependant, l'estimation est bruitée de part la nature bruitée des données.

En pratique, sur les machines Siemens du CHU de Nantes, l'estimation des coïncidences fortuites est réalisé avec la méthode de la fenêtre de retard. Des techniques de réduction de variance sont ensuite utilisées pour débruiter cette estimation [70].

Le sinogramme des coïncidences fortuites obtenu contribue au terme \bar{b} de l'équation (1.8).

1.2.1.2.4 Facteur de calibration Les images sont d'abord reconstruites en nombre de coïncidences détectées par voxel. En clinique, on souhaite obtenir une image avec une valeur plus interprétable, par exemple une concentration de radioactivité (en kBq/mL). En prenant un fantôme uniforme de radioactivité, et connaissant sa concentration initiale de radioactivité, il est possible de calculer un facteur de calibration associé au scanner TEP. Celui-ci permet d'obtenir une image en nombre de désintégrations émises par voxel. Finalement, en divisant par la durée d'acquisition, on peut obtenir des images en kBq/mL.

1.2.1.2.5 Autres corrections D'autres corrections doivent être mises en place pour être le plus proche de l'activité réelle dans le corps du patient. Sans être exhaustif, la correction de décroissance prend en compte la décroissance radioactive pendant le temps d'une acquisition. Après correction de la décroissance, l'image correspond alors à la concentration radioactive au début de l'acquisition. Quant à la correction du facteur d'embranchement, elle consiste à corriger le fait qu'un isotope puisse aussi se désintégrer selon d'autres modes que par désintégration β^+ .

En résumé, lors d'un examen TEP, un isotope radioactif est injecté dans le patient, puis se désintègre par émission β^+ . Le positon émis se déplace jusqu'à rencontrer un électron pour s'annihiler et donner naissance à deux photons gamma. Ceux-ci sont détectés par la machine TEP et sont ensuite transformés en photon lumineux qui sont amplifiés jusqu'à obtenir un signal électrique permettant de mettre en coïncidence deux photons détectés en un temps très court. Ces coïncidences sont enregistrées pour chaque paire de détecteurs. Le modèle direct relie mathématiquement ces données à l'image finale que l'on veut reconstruire. Pour cela, il utilise la géométrie du scanner TEP, mais prend aussi en compte sous forme de correction les phénomènes physiques affectant les données. Il s'agit maintenant de calculer l'image à partir des données, étape que l'on appelle reconstruction.

1.2.2 Le problème inverse de la reconstruction d'images TEP

Comme vu dans l'équation 1.8, le modèle direct TEP discrétisé s'écrit :

$$\bar{y} = Ax + \bar{b}$$

La matrice système et les contributions des événements diffusés et fortuits sont estimées en amont de la reconstruction (cf. partie 1.2.1.2). L'objectif de la reconstruction est de retrouver l'image x de la distribution radioactive du patient, à partir des données acquises et du modèle direct. On parle alors de problème inverse.

Pour résoudre ce type de problème, la solution la plus triviale serait d'essayer d'inverser la relation donnée par le modèle direct. Cela n'est pas possible pour le problème TEP car la matrice A n'est en général pas carrée (il y a plus de LOR que de voxels dans l'image) donc pas inversible. De plus, en choisissant un nombre de voxels et de LOR identiques, la matrice A serait trop grande pour être enregistrée en mémoire sur un ordinateur.

1.2.3 Reconstruction analytique

Des méthodes analytiques ont été développées dans un premier temps pour reconstruire des images TEP. La plus populaire est la rétroprojection filtrée (Filtered BackProjection - FBP), s'appuyant sur la transformée de Radon et la transformée de Fourier. En pratique, les reconstructions analytiques sont très rapides à effectuer, mais par nature extrêmement bruitées. En effet, les désintégrations radioactives surviennent de façon aléatoire, ce qui n'est pas modélisé par les méthodes analytiques.

Pour prendre en compte le caractère aléatoire des désintégrations radioactives, nous devons étudier quelle loi de probabilité régit les données. Nous pourrions alors la combiner à notre connaissance du modèle direct pour reconstruire une image.

1.2.4 Modélisation de la désintégration radioactive et de sa détection par la machine TEP

Considérons N_0 noyaux radioactifs. Le nombre de noyaux N qui se désintègrent pendant un temps t est proportionnel au nombre de noyaux qui ne se sont pas encore désintégrés :

$$\frac{dN}{dt} = \lambda_{\text{rad}} N(t)$$

λ_{rad} étant la constante de désintégration propre au noyau radioactif et reliée à sa demi-vie $t_{1/2}$ par la relation $\lambda_{\text{rad}} = \frac{\ln 2}{t_{1/2}}$. Le nombre total de noyaux qui ne se sont pas encore désintégrés à l'instant t est obtenu en résolvant l'équation différentielle précédente :

$$N(t) = N_0 e^{-\lambda_{\text{rad}} t}$$

Voyons d'abord la probabilité qu'un noyau radioactif se désintègre. D'après l'équation précédente, il y a $N_0 - N(t)$ noyaux qui se sont déjà désintégrés entre les temps 0 et t . Tous les noyaux ont la même probabilité de se désintégrer parmi les N_0 noyaux. Par conséquent, la probabilité p qu'un noyau se désintègre entre les temps 0 et t est la fraction de tous les noyaux désintégrés par rapport au nombre total de noyaux :

$$\pi = \frac{N_0 - N(t)}{N_0} = 1 - e^{-\lambda_{\text{rad}} t}$$

Par conséquent, la désintégration radioactive entre les temps 0 et t d'un de ces noyaux peut être modélisée par un processus de Bernoulli de paramètre π .

De plus, la probabilité qu'un noyau se désintègre exactement au temps t peut être trouvée en intégrant entre les temps 0 et t :

$$p(\text{désintégration du noyau } i \leq t) = \pi = 1 - e^{-\lambda_{\text{rad}} t} \quad (1.11)$$

$$= \int_0^t p(\text{désintégration du noyau } i = \tau) d\tau \quad (1.12)$$

Il s'ensuit que $p(\text{désintégration du noyau } i = \tau) = \lambda_{\text{rad}} e^{-\lambda_{\text{rad}} \tau}$ pour satisfaire l'égalité précédente. Ainsi, la durée de vie d'un noyau radioactif suit une loi exponentielle de paramètre λ_{rad} .

Nous pouvons maintenant étendre l'analyse à tous les noyaux. Toutes les désintégrations de noyaux sont des événements indépendants, et suivent la loi de Bernoulli. Ainsi, la probabilité que n noyaux se désintègrent entre les temps 0 et t suit une loi binomiale de paramètres $(n, 1 - e^{-\lambda_{\text{rad}} t})$.

Enfin, le nombre de désintégrations radioactives n qui se produisent pendant une acquisition entre les temps 0 et t est "élevé" (une injection de 100 MBq correspond à 10^8 désintégrations par seconde, donc $n > 30$), et nous supposons ici que la moyenne de la loi Binomiale $n\pi = n(1 - e^{-\lambda_{\text{rad}} t})$ est faible (nous supposons $n\pi < 15$ et $\pi < 0,1$). Par conséquent, en utilisant la loi des événements rares, la loi binomiale peut être approximée par une loi de Poisson de paramètre $\lambda_{\text{desint}} = n\pi$. Ainsi, la probabilité que n noyaux radioactifs se désintègrent en même temps est donnée par :

$$p(n \text{ désintégrations}) = \frac{e^{-n\pi} (n\pi)^n}{n!}$$

L'hypothèse précédente, à savoir que $n\pi$ est faible, n'est pas toujours valable pour tous les radiotraceurs [85]. Cependant, en pratique, les détecteurs ont une efficacité de détection ε faible. Toutes les dérivations précédentes pourraient donc être remplacées par la prise en compte de la désintégration d'un événement, conjointement à sa détection par les détecteurs TEP. En utilisant la sélection binomiale d'une loi de Poisson [6], la détection de n désintégrations pourraient encore être approximées par la loi de Poisson de paramètre $\varepsilon \lambda_{\text{desint}} = \varepsilon n\pi$:

$$p(n \text{ désintégrations}) = \frac{e^{-\varepsilon n\pi} (\varepsilon n\pi)^n}{n!}$$

Comme l'efficacité de détection ε est faible, les hypothèses pour utiliser la loi des événements rares deviennent valables pour l'imagerie TEP [85].

Le tableau 1.3 résume les propriétés mathématiques des désintégrations radioactives, de la loi de Bernoulli jusqu'à la loi de Poisson.

Propriétés des désintégrations	Désintégration de 1 noyau	Bernoulli $\pi = 1 - \exp(-\lambda_{\text{rad}}t)$
	Durée de vie de 1 noyau	Exponentielle (λ_{rad})
	Désintégration de n noyaux	Binomiale (n, π)
	Désintégration de n noyaux si $n \gg 1$ désintégrations et si <ul style="list-style-type: none"> • $n\pi < 15$, • $\pi < 0,1$ 	Poisson(λ_{desint}) $\lambda_{\text{desint}} = \varepsilon n \pi$
Propriété des détections de désintégrations	Désintégration de n noyaux si $n \gg 1$ désintégrations + faible efficacité des détecteurs TEP en pratique	Poisson($\varepsilon \lambda_{\text{desint}}$)

TABLE 1.3 – Tableau récapitulatif des propriétés probabilistes des désintégrations radioactives

1.2.5 Reconstruction de données en format sinogramme par méthodes itératives

1.2.5.1 Vraisemblance de Poisson

Pour résoudre le problème inverse TEP en prenant en compte le côté probabiliste des données, une idée est de chercher une solution x la plus "proche" de la distribution radioactive réelle, étant données les données acquises. Au sens probabiliste, cela revient à maximiser la probabilité d'obtenir les données acquises connaissant la distribution radioactive x dans le patient, probabilité que l'on appelle la vraisemblance (likelihood en anglais).

Soit Y_i la variable aléatoire comptant le nombre de coïncidences dans la LOR i , qui sont la conséquence des désintégrations radioactives assignées par le système TEP à la LOR i . Soit

la variable aléatoire $Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_I \end{bmatrix}$ contenant les I variables aléatoires Y_i de chaque LOR i , de

réalisation $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_I \end{bmatrix}$. La distribution radioactive x dans le patient est une réalisation d'une

variable aléatoire que l'on note X . La vraisemblance $p(Y | X)$ est une fonction de l'image x et des données dont on a une réalisation y fixe : $x \mapsto p(Y | X)(y, x)$.

Ainsi, d'après le tableau 1.3, la variable aléatoire $Y_i | X$ suit une loi de Poisson. La probabilité d'avoir y_i coïncidences dans la LOR i , sachant que la distribution radioactive a pris la valeur x est donc donnée par :

$$p(Y_i = y_i | X = x) = \frac{e^{-\bar{y}_i} \bar{y}_i^{y_i}}{y_i!}$$

avec $\bar{y}_i = E(Y_i | X = x)$. Par souci de simplicité, nous noterons $p(Y_i = y_i | X = x)$ comme $p(Y_i = y_i | x)$, ou de façon plus compacte $p(y_i | x)$.

Le nombre de coïncidences sur chaque LOR est indépendant car provient de désintégrations radioactives indépendantes, ce qui permet d'écrire la vraisemblance pour l'ensemble des LOR :

$$p(Y = y | X = x) = \prod_{i=1}^I p(Y_i = y_i | X = x) = \prod_{i=1}^I \frac{e^{-\bar{y}_i} \bar{y}_i^{y_i}}{y_i!}$$

1.2.5.2 Problème d'optimisation du maximum de vraisemblance

On cherche à maximiser cette vraisemblance, ce qui revient à résoudre le problème d'optimisation suivant :

$$\hat{x} = \arg \max_x p(Y = y | X = x) \quad (1.13)$$

Une astuce pour résoudre plus simplement le problème d'optimisation (1.13) est de passer au logarithme (le logarithme étant une fonction croissante). On obtient alors l'expression de la log-vraisemblance (notée $L(y | x)$) en remplaçant l'expression du modèle direct :

$$L(y | x) = \ln p(y | x) \quad (1.14)$$

$$= \sum_{i=1}^I (-\bar{y}_i + y_i \ln(\bar{y}_i) - \ln(y_i)) \quad (1.15)$$

$$= \sum_{i=1}^I \left(y_i \ln \left(\sum_{k=1}^p A_{ik} x_k + \bar{b}_i \right) - \left(\sum_{k=1}^p A_{ik} x_k + \bar{b}_i \right) - \ln(y_i) \right) \quad (1.16)$$

$$= y^T \ln(Ax + \bar{b}) - 1^T(Ax + \bar{b}) - 1^T \ln(y!) \quad (1.17)$$

Le problème d'optimisation revient donc à maximiser la log-vraisemblance en négligeant le terme constant, ce qui donne :

$$\hat{x} = \arg \max_x \Phi(x) \quad (1.18)$$

$$= \arg \max_x y^T \ln(Ax + \bar{b}) - 1^T(Ax + \bar{b}) \quad (1.19)$$

Φ est appelée fonction de coût, c'est-à-dire la fonction que l'on cherche à optimiser. Pour résoudre ce problème d'optimisation de façon analytique, il faut calculer la dérivée seconde de la log-vraisemblance pour étudier sa concavité (ce qui donnera l'existence de la solution) et chercher x qui annule sa dérivée. En calculant la dérivée première puis la dérivée seconde, on obtient :

$$\frac{\partial L(y | x)}{\partial x_j} = \sum_{i=1}^I A_{ij} \left(\frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i} - 1 \right) \quad (1.20)$$

$$\frac{\partial^2 L(y | x)}{\partial x_j \partial x_l} = - \sum_{i=1}^I \frac{A_{ij} A_{il} y_i}{\left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right)^2} \quad (1.21)$$

On a bien $\frac{\partial^2 L(y|x)}{\partial x_j \partial x_l} \leq 0$ donc il existe une solution au problème d'optimisation (1.19)

Pour déterminer les possibles solutions, il faut trouver les valeurs de \hat{x} annulant la dérivée première. Mais résoudre ce problème analytiquement n'est pas possible.

Ainsi, pour prendre en compte la variabilité statistique des données, on peut dériver la vraisemblance liée au problème TEP. Mais il est ensuite nécessaire de dériver des méthodes itératives pour maximiser la log-vraisemblance. Avant de discuter des méthodes itératives développées en TEP, nous allons voir que chercher le maximum de vraisemblance induit des images trop bruitées, et que des solutions sont nécessaires pour pouvoir exploiter visuellement les images.

1.2.5.3 Nature mal posée du problème TEP et solutions possibles

Le problème inverse TEP est double : d'une part la relation pour obtenir l'image x n'est pas directe ; d'autre part le problème est mal posé. Cela signifie qu'une faible variation sur les données TEP acquises peut entraîner une grande différence sur les images reconstruites. En pratique, les données TEP ne possèdent qu'une quantité finie de coïncidences détectées dans chaque LOR, ce qui entraîne une amplification des différences dans chaque voxel de l'image reconstruite et donc du bruit dans l'image.

Dans le problème inverse TEP, chercher le maximum de vraisemblance (MLE - Maximum Likelihood Estimation) ne prend en compte que l'adéquation aux données pour reconstruire une image TEP. La solution donnée par le maximum de vraisemblance est donc théoriquement très bruitée, ce qui est aussi vérifié en pratique et rend son utilisation impossible. Il existe plusieurs solutions pour limiter la quantité de bruit dans l'image reconstruite :

- La première est de stopper avant convergence les algorithmes ML (Maximum Likelihood - faisant du MLE). En effet, à partir d'un certain nombre d'itérations, ce type d'algorithme ne diminue plus le biais de l'image reconstruite par rapport à l'image théorique, mais augmente seulement le bruit. La difficulté est de savoir à combien d'itérations stopper l'algorithme.
- La deuxième est de faire du MLE, puis d'appliquer une méthode de débruitage sur l'image reconstruite bruitée. En TEP, le filtrage gaussien est souvent utilisé.
- La troisième est d'ajouter une fonction de régularisation R , aussi appelé pénalité, que nous allons détailler dans la partie suivante.

1.2.5.4 Ajout d'un terme de régularisation à la vraisemblance

Une solution pour limiter la quantité de bruit dans l'image est d'imposer une pénalité sur l'image à reconstruire. Une façon de faire est de se placer dans un cadre bayésien en considérant une distribution de probabilité a priori $p(x)$ sur l'image reconstruite x . Pour utiliser cette connaissance a priori sur l'image x , on peut partir de la vraisemblance et utiliser la relation de Bayes :

$$p(x | y) = \frac{p(x)p(y | x)}{p(y)} \quad (1.22)$$

La formulation bayésienne conduit à chercher une image x qui maximise la distribution a posteriori $p(x | y)$, ce qu'on appelle résoudre un problème MAP (Maximum A Posteriori). Mathématiquement, cela s'écrit :

$$\hat{x} = \arg \max_x p(x | y) \quad (1.23)$$

Comme pour la vraisemblance, on peut passer au logarithme et retirer les constantes ne dépendant pas de x :

$$\hat{x} = \arg \max_x p(x | y) \quad (1.24)$$

$$= \arg \max_x \frac{p(x)p(y | x)}{p(y)} \quad (1.25)$$

$$= \arg \max_x \ln \left(\frac{p(x)p(y | x)}{p(y)} \right) \quad (1.26)$$

$$= \arg \max_x \ln(p(y | x) + \ln(p(x)) - \ln(p(y))) \quad (1.27)$$

$$= \arg \max_x L(y | x) + \ln(p(x)) \quad (1.28)$$

$$(1.29)$$

On peut poser $p(x) = \frac{1}{Z} \exp(-\beta R(x))$, ce qui n'impose rien sur le choix de la distribution a priori (Z est choisi pour que l'intégrale de cette densité soit égale à 1), ce qui donne :

$$\hat{x} = \arg \max_x L(y | x) - \beta R(x) \quad (1.30)$$

Le problème MAP peut se voir alors comme un problème d'optimisation de la vraisemblance auquel on a ajouté une fonction de régularisation R , dont la somme est pondérée par un paramètre de pénalité β . Comme $p(x)$ représente la distribution a priori sur l'image x , la fonction R peut être vue comme une façon de pénaliser des images qui seraient non désirables. Plusieurs fonctions R ont été proposées dans la littérature [71]. Nous présenterons ici celles qui ont été utilisées pendant la thèse. Nous parlerons dorénavant de problème PML (Penalized Maximum Likelihood) plutôt que de problème MAP pour caractériser l'équation 1.30 car certaines fonctions R proposées dans la littérature ne sont pas issues d'une distribution de probabilité $p(x)$ [71].

Nous allons maintenant voir les équations de plusieurs algorithmes de reconstruction, utilisant ou non des termes de pénalité (respectivement des algorithmes ML ou PML). Ensuite, nous verrons plus en détail des exemples de pénalité utilisées dans ce travail.

1.2.5.5 Méthodes itératives cliniques

De nombreuses méthodes ont été développées et discutées dans [71], qui relate l'évolution des méthodes jusqu'en 2006. Nous allons en voir plusieurs d'entre elles qui sont encore aujourd'hui utilisées sur les machines en clinique : MLEM [82], accéléré avec des sous-ensembles par la méthode OSEM [52] et utilisée notamment par Siemens Healthcare, ainsi que BSREM utilisée par GE Healthcare.

1.2.5.5.1 Algorithme MLEM L'algorithme Maximum Likelihood Expectation Maximization (MLEM) [82] a été développé en s'appuyant sur l'algorithme plus général d'espérance maximisation (Expectation Maximization - EM). Cet algorithme cherche à maximiser une fonction (ici la vraisemblance) de façon itérative en deux étapes : l'étape E et l'étape M. L'étape E consiste à estimer les données "latentes" (variable aléatoire notée Z), qui sont des données complètes mais non observées en pratique. Cette estimation est faite à partir des données TEP observées et de l'estimation courante $x^{(k)}$ des paramètres d'intérêt (ici les voxels de l'image x). Ensuite pendant l'étape M, l'idée est de maximiser une fonction que l'on sait optimiser. Les données latentes sont alors choisies de sorte à obtenir une fonction que l'on sait optimiser [9]. Cette fonction est appelée fonction de substitution (surrogate function) car elle a pour but de remplacer la fonction que l'on veut optimiser initialement.

Après dérivation (cf. [9] pour la dérivation mathématique), l'algorithme EM consiste à maximiser de façon itérative l'espérance conditionnelle de la log-vraisemblance des données latentes [71] :

- étape E : Définir la fonction $\Phi'(x, x^{(k)}) = \mathbb{E} [L(Z | x) | (y; x^{(k)})]$
- étape M : Calculer $x^{(k+1)} = \arg \max \Phi'(x, x^{(k)})$

Pour le problème TEP, on obtient l'expression des voxels x_j de l'algorithme MLEM calculés lors de l'étape M avec l'équation 1.31 :

$$x_j^{n+1} = x_j^n + \frac{x_j^n}{\sum_{i=1}^I A_{ij}} \frac{\partial}{\partial x_j} \Phi(x^n) = x_j^n + \frac{x_j^n}{\sum_{i=1}^I A_{ij}} \sum_{i=1}^I A_{ij} \left(\frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i} - 1 \right) \quad (1.31)$$

avec J le nombre de voxels dans l'image reconstruite x .

Les itérés de l'algorithme MLEM dans l'équation (1.31), écrits sous forme "additive". De cette façon, on voit que l'algorithme MLEM n'est qu'un algorithme de descente de gradient, avec une suite de pas α_n identiques et égaux à 1, et un préconditionneur propre à chaque voxel $d_j(x^n) = \frac{x_j^n}{\sum_{i=1}^I A_{ij}}$. L'algorithme MLEM peut aussi s'écrire de façon "multiplicative" en réorganisant les termes comme dans l'expression 1.32 :

$$x_j^{n+1} = \frac{x_j^n}{\sum_{i=1}^I A_{ij}} \sum_{i=1}^I A_{ij} \frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i} \quad (1.32)$$

Cet algorithme ne nécessite que peu d'opérations coûteuses (projection et rétroprojection à chaque itération, calcul de l'image de sensibilité $\left(\sum_{i=1}^I A_{ij}\right)_j$ une seule fois). En revanche, cet algorithme n'est jamais utilisé jusqu'à convergence car il produit des images beaucoup trop bruitées. En pratique, on peut le lancer avec un grand nombre d'itérations, puis appliquer un filtre gaussien pour débruiter l'image reconstruite.

L'algorithme MLEM peut être accéléré par la méthode des sous-ensembles (subsets en anglais), ce qui a conduit au développement de la méthode OSEM.

1.2.5.5.2 Algorithme OSEM Pour accélérer la reconstruction, on peut diviser la fonction de vraisemblance Φ en M sous-fonctions Φ_m . Chaque sous-fonction Φ_m a pour but de ne prendre en compte les données que sur un sous-ensemble S_m de l'ensemble de toutes les LOR [52]. La fonction de coût peut se décomposer de la façon suivante :

$$\Phi(x) = \sum_{i=1}^I \left[y_i \ln \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) - \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) \right] \quad (1.33)$$

$$= \sum_{m=1}^M \left[\sum_{i \in S_m} y_i \ln \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) - \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) \right], \quad \text{avec } \bigcup_{m=1}^M S_m = \{1, 2, \dots, I\} \quad (1.34)$$

$$= \sum_{m=1}^M \Phi_m(x) \quad (1.35)$$

On remarque alors que les sous-fonctions Φ_m sont exactement les mêmes que la fonction Φ , seul les indices i des LOR parcourues change :

$$\Phi_m(x) = \sum_{i \in S_m} \left(y_i \ln \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) - \left(\sum_{k=1}^J A_{ik} x_k + \bar{b}_i \right) - \ln(y_i) \right) \quad (1.36)$$

Les itérés de chaque sous-ensemble se dérivent ensuite de la même façon que pour l'algorithme MLEM, et sont traités séquentiellement du premier sous-ensemble au dernier (d'où le nom Ordered Subsets EM pour l'algorithme OSEM) :

$$\begin{cases} x_j^{n,m} = x_j^{n,m-1} + \frac{x_j^{n,m-1}}{\sum_{i \in S_m} A_{ij}} \frac{\partial}{\partial x_j} \Phi_m(x^n), & \text{mise à jour intra sous-ensemble} \\ x_j^{n+1} = x_j^{n,M}, & \text{mise à jour à chaque itération} \end{cases} \quad (1.37)$$

De la même façon, chaque mise à jour des voxels est de la même forme qu'une descente de gradient, avec une suite de pas α_n identiques et égaux à 1, un préconditionneur propre à chaque voxel et qui dépend du sous-ensemble considéré :

$$d_j^m(x^n) = \frac{x_j^n}{\sum_{i \in S_m} A_{ij}} \quad (1.38)$$

En revanche, comme montré sur la figure 1.15, l'algorithme OSEM ne converge pas complètement vers le maximum de vraisemblance, et entre dans un cycle après plusieurs itérations. Mais

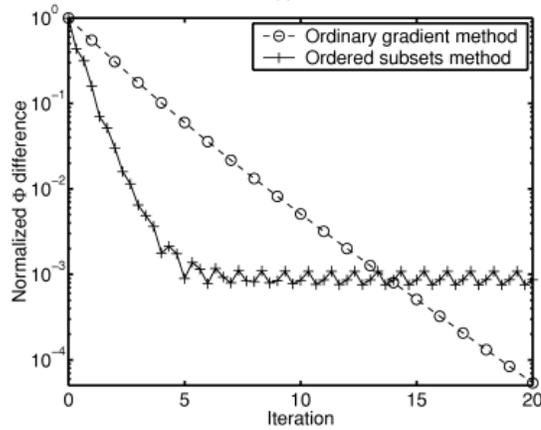


FIGURE 1.15 – Convergence d’algorithme utilisant des sous-ensembles (comme OSEM) par rapport au même algorithme n’en utilisant pas (comme MLEM). La figure est extraite de l’article [1].

étant donné sa convergence beaucoup plus rapide après seulement quelques itérations comparé à MLEM (cf. figure 1.15), il lui est préféré en clinique.

Le nombre de sous-ensembles M est défini en amont de la reconstruction, et les sous-ensembles sont choisis de façon déterministe. L’algorithme est habituellement lancé avec 2 itérations et un grand nombre de sous-ensembles (28 pour le Siemens Biograph mMR, 21 pour le Siemens Biograph mCT), avec un post filtrage gaussien. Le nombre d’itérations peut être un tout petit peu plus élevé en fonction du type d’acquisition et d’application TEP. Ainsi, l’algorithme n’est pas amené à convergence, mais on considère en pratique que cela suffit pour avoir un biais raisonnable dans l’image. Les itérations supplémentaires ne font qu’augmenter le bruit dans l’image.

1.2.5.5.3 Algorithme BSREM L’algorithme Block Sequential Regularized Expectation Maximization (BSREM) a été proposé pour deux raisons par rapport à OSEM :

- obtenir un algorithme utilisant des sous-ensembles mais convergeant globalement et non stagnant dans un cycle,
- pouvoir prendre en compte une fonction de régularisation pour limiter la quantité de bruit dans l’image reconstruite.

L’algorithme BSREM a d’abord été proposé par [27], puis amélioré par [1] pour établir la preuve de convergence avec des hypothèses plus réalistes. Il a été imaginé en partant de la forme générale d’un algorithme de descente de gradient :

$$x_j^{n,m} = x_j^{n,m-1} + \alpha_n d_j(x^{n,m-1}) \frac{\partial}{\partial x_j} \Phi(x^{n,m-1}) \quad (1.39)$$

avec α_n une suite de pas et $d_j(x^n)$ une fonction de mise à l’échelle non négative, qui agit comme un préconditionneur.

La première idée de BSREM est de faire de la relaxation à partir de la suite de pas α_n , c’est-à-dire d’utiliser une suite de pas α_n décroissante, ce qui empêcherait l’algorithme de rester coincé dans un cycle comme OSEM (où $\alpha_n = 1$). [1] donne une condition mathématique suffisante sur la suite choisie pour démontrer la convergence de l’algorithme :

$$\left\{ \begin{array}{l} \sum_{k=0}^{\infty} \alpha_n = \infty \\ \sum_{k=0}^{\infty} \alpha_n^2 < \infty \end{array} \right. \quad (1.40)$$

Une suite qui vérifie ces propriétés a été proposée dans [1] avec l'expression suivante :

$$\alpha_n = \frac{\alpha_0}{\gamma_B n + 1}, \forall n \in \mathbb{N} \quad (1.41)$$

avec $\alpha_0 > 0$ le pas initial, et $\gamma_B > 0$ un paramètre pour régler la rapidité de décroissance de la suite. A ce jour, il n'existe pas de méthode pour choisir ces 2 paramètres de façon optimale.

En outre, pour obtenir la convergence globale de l'algorithme vers l'optimum désiré, l'algorithme BSREM utilise aussi un préconditionneur qui ne dépend pas du sous-ensemble considéré, contrairement à OSEM (cf. (1.38)). En effet, les auteurs dans [1] ont montré qu'OSEM peut converger si on applique une relaxation des pas α_n , mais vers un autre point que l'extremum désiré, à cause du préconditionneur qui dépend du sous-ensemble considéré.

Premièrement, le préconditionneur choisi par [27] était $d_j^Y(x^n) = x_j^n$. Ainsi, les hypothèses de convergence étaient :

- chaque image x^n à l'itération n de l'algorithme doit être positive et bornée : $\forall n, x^n \in \mathcal{B} = \{x^n \in \mathbb{R}^J, \forall j \in [1; J], 0 \leq x_j^n \leq U\}$,
- la suite $(\Phi(x^n))_k$ converge.

Dans [1], le préconditionneur a été remplacé par $d_j^A(x^n) = \begin{cases} \frac{x_j^n}{\sum_{i=1}^I A_{ij}}, & \text{si } 0 \leq x_j^n < \frac{U}{2} \\ \frac{U - x_j^n}{\sum_{i=1}^I A_{ij}}, & \text{si } \frac{U}{2} \leq x_j^n \leq U \end{cases}$.

Ainsi, les hypothèses de convergence sont devenues plus précises :

- les conditions données par les équations (1.40), qui sont vérifiées par exemple par la suite proposée dans l'équation (1.41),
- $\forall n \in \mathbb{N}, x^n \in \mathcal{B}^O$ (l'intérieur de l'ensemble \mathcal{B}). Cela est vérifié si plus précisément si :
 - $x_0 \in \mathcal{B}^O$,
 - la suite de pas α_n est bornée par une constante t "suffisamment petite".

Les auteurs dans [1] ont remarqué que $\alpha_n \xrightarrow[k \rightarrow \infty]{} 0$ d'après les hypothèses dans (1.40). Par conséquent, $\exists N \in \mathbb{N}, \forall n \geq N, \alpha_n < t$. Leur idée est d'ajouter une étape supplémentaire pour assurer que $\forall n \in \mathbb{N}, x^n \in \mathcal{B}^O$, et en particulier alors $x^N \in \mathcal{B}^O$, qui serait vu comme le point initial x_0 pour la preuve de convergence.

Pour cela, après passage par tous les ensembles et mise à jour de l'image x dans (1.39), l'itéré est projeté dans $\mathcal{I} = \{x \in \mathbb{R}^J, \forall j \in [1; J], t \leq x_j \leq U - t\}$ pour conserver l'image x dans \mathcal{B}^O :

$$x_j^n = \begin{cases} t, & \text{si } x_j^n \leq 0 \\ U - t, & \text{si } x_j^n \geq U \\ x_j^n, & \text{si } x_j^n \in]0, U[\end{cases} \quad (1.42)$$

Enfin, l'extension de l'algorithme avec pénalité se fait directement, en répartissant la fonction de régularisation R sur les M sous-fonctions de coût Φ_m , par exemple :

$$\Phi_m(x) = \Phi_m^{\text{OSEM}}(x) - \beta \frac{1}{M} R(x) \quad (1.43)$$

La preuve de convergence reste valide tant que la fonction de régularisation choisie est strictement concave.

Dans toute la suite, nous parlerons de l'algorithme modifié proposé dans [1] en utilisant l'acronyme BSREM.

1.2.5.5.4 Contrainte intrinsèque de non-négativité Les algorithmes présentés dans la partie précédente ont une contrainte intrinsèque de non-négativité. Cela veut dire que les voxels de l'image restent positifs du moment où l'initialisation est choisie positive. Ce type d'algorithme sont non biaisés uniquement asymptotiquement, c'est-à-dire pour un nombre de coïncidences tendant vers l'infini et une reconstruction poussée à convergence. Plus le bruit est important, plus des zones peuvent devenir biaisés. En effet, prenons un objet dont on veut faire

une image où il n'y a pas d'activité dans une région de l'objet. Les voxels de l'image dans cette région doivent donc être mis à zéro par l'algorithme de reconstruction pour être fidèles à la distribution radioactive réelle. Or, tous les algorithmes présentent un certain niveau de bruit dans l'image dû à la nature mal posée du problème inverse TEP. Ces voxels n'auront donc pas tous la même valeur, et seront donc strictement positifs ou nuls. Cette région froide est donc reconstruite avec un biais positif par les algorithmes ayant une contrainte intrinsèque de non-négativité. Une explication plus mathématique liée à cette observation est présentée pour un problème inverse simplifié dans [10].

En particulier, dans certaines applications avec peu de statistiques d'acquisition, les images reconstruites par les algorithmes précédents sont connues pour produire un biais positif dans les régions froides (c'est-à-dire les régions de faible activité relative par rapport aux autres régions) [16].

Les méthodes que nous allons voir dans la prochaine partie ont été proposées spécifiquement pour enlever la contrainte de non-négativité dans l'image et réduire le biais.

1.2.5.6 Méthodes itératives pour réduction de biais

Plusieurs algorithmes de reconstruction autorisant des voxels négatifs ont été développés pour la reconstruction TEP, visant à réduire le biais positif dans les régions froides : ABML [13], NEGML [66] et leurs versions modifiées [93], ADMM-Reg [59], APPGML [62], HypoC-PML [10]. Dans cette partie, nous allons voir en détail les algorithmes APPGML, s'appuyant sur l'algorithme AML et PPGML, ainsi que l'algorithme ADMM-Reg.

1.2.5.6.1.0 Algorithme PPGML *NB* : Cet algorithme n'est pas un algorithme ayant pour but de réduire le biais dans les régions froides. Il est présenté ici pour comprendre la dérivation de l'algorithme APPGML 1.2.5.6.3.

L'algorithme Penalized Preconditioned Gradient Maximum Likelihood (PPGML) a été proposé par [65]. Il s'appuie sur l'algorithme One Step Late (OSL) implémenté par [40], qui ajoute un terme de pénalité à partir de l'écriture multiplicative de MLEM. Quant à PPGML, il introduit le terme de pénalité en repartant de l'écriture additive de l'algorithme MLEM. Voici l'équation de mise à jour des itérés de OSL :

$$x_j^{n+1} = \frac{x_j^n}{\sum_{i=1}^I A_{ij} - \beta \frac{\partial R(x^n)}{\partial x_j}} \sum_{i=1}^I A_{ij} \frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i} \quad (1.44)$$

Et pour PPGML :

$$x_j^{n+1} = x_j^n + \frac{\frac{\partial}{\partial x_j} \Phi(x^n)}{\frac{\sum_{i=1}^I A_{ij}}{x_j^n} - \frac{\partial^2}{\partial x_j^2} \Phi(x^n)} \quad (1.45)$$

$$= x_j^n + \frac{\sum_{i=1}^I A_{ij} \left(\frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i} - 1 \right) - \beta \frac{\partial}{\partial x_j} R(x^n)}{\frac{\sum_{i=1}^I A_{ij}}{x_j^n} - \beta \frac{\partial^2}{\partial x_j^2} R(x^n)} \quad (1.46)$$

L'algorithme PPGML a été développé pour une meilleure stabilité numérique que OSL. Il n'a pas de preuve de convergence, mais celle-ci est tout de même observée en pratique [65].

1.2.5.6.2 Algorithme AML *NB* : Pour cette partie et la suivante, nous noterons la matrice système avec la lettre C , ainsi que les éléments de cette matrice c_{ij} pour éviter toute confusion avec la borne A de l'algorithme AML.

L'algorithme ABML ("A"-B" Maximum Likelihood) a d'abord été introduit dans [13]. ABML cherche à minimiser la divergence de Kullback-Leibler (KL-divergence) entre les données acquises y et le modèle direct $\bar{y} = Cx + \bar{b}$, ce qui est équivalent à maximiser la log-vraisemblance de Poisson (les deux fonctions sont les mêmes à une constante près, ce qui n'affecte pas l'optimisation) [71]. On introduit deux variables $A_{\text{all}} = (A_j)_{j \in [1; J]}$ et $B_{\text{all}} = (B_j)_{j \in [1; J]}$ qui servent à borner les voxels x_j obtenus pendant la résolution du problème d'optimisation. On comprend alors qu'en choisissant des valeurs de A_j négatives, la solution x estimée peut alors contenir des voxels négatifs.

Cet algorithme a ensuite été utilisé en TEP. Il a démontré une aptitude à réduire le biais en particulier dans les zones froides [30].

L'algorithme AML ("A" Maximum Likelihood) est un sous cas de l'algorithme ABML, où on choisit $\forall j \in [1; J], B_j = \infty$ pour ne pas borner les valeurs les plus élevées, et $\forall j \in [1; J], A_j = A$, avec $A \in \mathbb{R}$ [93]. L'équation de mise à jour des voxels est donné par l'équation 1.47 :

$$x_j^{n+1} - A = \frac{x_j^n - A}{\sum_{i=1}^I c_{ij}} \sum_{i=1}^I c_{ij} \frac{y_i - A \sum_{k=1}^J c_{ik}}{\bar{y}_i - A \sum_{k=1}^J c_{ik}} \quad (1.47)$$

On voit la correspondance avec l'équation de l'algorithme MLEM 1.32 : l'équation d'AML 1.47 est exactement la même, mais où les voxels de l'image ont été décalés de $-A$, et les données ainsi que leur moyenne ont été décalées de $-A \sum_{k=1}^J c_{ik}$. Ainsi, les voxels x_j ne peuvent pas descendre en dessous de la borne A par construction.

1.2.5.6.3 Algorithme APPGML L'algorithme APPGML est une combinaison de l'idée d'AML de décaler l'image et les données, appliqué à un algorithme pouvant prendre en compte une pénalité [62]. L'algorithme PPGML a été choisi pour sa stabilité numérique. En décalant l'image d'un facteur A , et les données et leur moyenne d'un facteur $-A \sum_{k=1}^J c_{ik}$, l'algorithme APPGML s'écrit comme suit :

$$x_j^{n+1} = x_j^n + \frac{\sum_{i=1}^I A_{ij} \left(\frac{y_i - A \sum_{k=1}^J c_{ik}}{\bar{y}_i - A \sum_{k=1}^J c_{ik}} - 1 \right) - \beta \frac{\partial}{\partial x_j} R(x^n)}{\sum_{i=1}^I c_{ij} - \beta \frac{\partial^2}{\partial x_j^2} R(x^n)} \quad (1.48)$$

De cette façon, APPGML est bornée par la borne A , qui peut être choisie négative tout comme AML pour autoriser les valeurs négatives, et ainsi apporter une réduction de biais. APPGML a comme avantage sur AML de pouvoir prendre en compte une pénalité [62].

1.2.5.6.4 Algorithme ADMM-Reg *NB* : Nous reprenons la notation initiale A pour la matrice système.

L'algorithme ADMM-Reg est un algorithme aussi développé pour obtenir une réduction de biais dans les régions froide en autorisant les valeurs négatives et en retirant toute contrainte intrinsèque de non-négativité sur l'image x . Pour cela, l'algorithme ADMM-Reg (présenté dans un premier temps sans terme de régularisation) cherche à minimiser la log-vraisemblance négative de Poisson en ajoutant une contrainte de non négativité dans l'espace des projections [59] :

$$\begin{cases} \min_x -L(y|x) \\ Ax + \bar{b} > 0 \end{cases} \quad (1.49)$$

Pour dériver l'ADMM-Reg, les auteurs de [59] ont d'abord utilisé une fonction barrière g^{Lim} définie comme suit³ :

$$g^{Lim}(\eta) = \sum_i g^{Lim}(\eta_i), \quad g^{Lim}(\eta_i) = \begin{cases} \infty, & y_i > 0, \eta_i < 0 \\ 0, & \text{else} \end{cases}$$

Ainsi, la contrainte peut être retirée en ajoutant la fonction barrière dans la fonction à optimiser :

$$(1.49) \iff \min_x L(y|x) + g^{Lim}(Ax + \bar{b}) \quad (1.50)$$

Les auteurs ont ensuite réécrit le problème d'optimisation (1.50) en utilisant une variable auxiliaire $v = Ax$. Ainsi, on cherche à résoudre un problème d'optimisation sur les variables v et x avec une contrainte linéaire liant ces deux variables. L'algorithme ADMM (Alternating Direction Method of Multipliers [12] (cf. Annexe B.1 pour plus de détails) a été utilisé pour dériver la méthode ADMM-Reg [59]. La mise à jour des voxels de l'image x alterne avec la mise à jour d'un sinogramme v et d'un sinogramme u par le système suivant :

$$x^{n+1} = \arg \min_x \frac{\rho_2}{2} \|Ax - v^n + u^n\|_2^2 \quad (1.51a)$$

$$v^{n+1} = \arg \min_v 1^T(v + \bar{b}) - y^T \ln(v + \bar{b}) + g^{Lim}(v + \bar{b}) + \frac{\rho_2}{2} \|Ax^{n+1} - v + u^n\|_2^2 \quad (1.51b)$$

$$u^{n+1} = u^n + Ax^{n+1} - v^{n+1} \quad (1.51c)$$

où ρ_2 est le paramètre de pénalité ADMM.

Il est possible d'ajouter une fonction de régularisation R au problème d'optimisation (1.49), ce qui se répercute seulement dans l'équation (1.51a) qui devient alors :

$$x^{n+1} = \arg \min_x \frac{\rho_2}{2} \|Ax - v^n + u^n\|_2^2 + \beta R(x) \quad (1.52)$$

Nous ne détaillerons que dans la partie 2.3.1 les équations pour résoudre les équations (1.51a), (1.52) et (1.51b) qui sont communes avec l'algorithme développé dans le chapitre suivant (cf. partie 2.3.2).

1.2.5.7 Fonctions de régularisation

L'ajout d'une fonction de régularisation, ou pénalité, permet d'optimiser à la fois selon les données (la vraisemblance) et selon une information supplémentaire sur l'image. Étudions maintenant plusieurs fonctions de régularisation proposées dans la littérature et qui sont utiles pour cette thèse.

1.2.5.7.1 Pénalité quadratique Une façon possible pour limiter le bruit dans l'image est de choisir des fonctions qui pénalisent les valeurs élevées des voxels dans l'image x , par exemple la fonction quadratique :

$$R : x \in \mathbb{R}^J \mapsto \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_J^2 \end{bmatrix} \quad (1.53)$$

Cette pénalité peut chercher à s'approcher d'une image de référence x^{ref} en adaptant la fonction R comme suit :

$$R : x \in \mathbb{R}^J \mapsto \begin{bmatrix} (x_1 - x_1^{ref})^2 \\ (x_2 - x_2^{ref})^2 \\ \vdots \\ (x_J - x_J^{ref})^2 \end{bmatrix} \quad (1.54)$$

3. Les équations sont présentées de la même façon que dans le papier original [59]. Les discordances dans les symboles de comparaison sont discutées dans l'annexe B.3

Pour représenter plus finement une image qui aurait du sens visuellement, la fonction de régularisation peut prendre en compte une information spatiale. Dans la suite, nous allons voir en détail les fonctions de régularisation de type Markov Random Field (MRF) qui utilisent de l'information spatiale.

1.2.5.7.2 Pénalité MRF Un champ aléatoire markovien (Markov Random Field (MRF) en anglais) [35] correspond à un ensemble de variables aléatoires $(X_j)_{j=1}^J$ dont la relation est décrite par un graphe non orienté. Les arêtes du graphe représentent les relations de dépendance entre les variables aléatoires. La loi conjointe de l'ensemble des variables aléatoires du graphe doit vérifier la propriété de Markov : une variable aléatoire X_j dépend uniquement des valeurs des variables aléatoires qui partagent une arête en commun avec elle, et pas des autres variables aléatoires du graphe.

Dans le cas de l'imagerie TEP, chaque voxel est un noeud d'un graphe, considéré comme une variable aléatoire. Les arêtes du graphe représentent les relations de voisinage entre voxels : pour un voxel x_j , on définit un voisinage \mathcal{N}_j comme l'ensemble des valeurs de k tel que le voxel x_k lui soit "suffisamment proche" selon une distance. Par exemple, pour des voxels de coordonnées X_j et X_k on peut définir un voisinage comme une sphère de rayon r , selon la distance euclidienne : $\mathcal{N}_j = \{k \in \mathbb{N}, \|X_j - X_k\|_2^2 < r\}$.

On peut alors écrire la formule générale de pénalité MRF utilisée en TEP⁴ :

$$R(x) = \sum_{j=1}^J \sum_{k \in \mathcal{N}_j} w_{jk} \Psi(x_j - x_k) \quad (1.55)$$

Pour chaque voxel x_j , on regarde l'écart entre ce voxel et tous les voxels de son voisinage, à travers une fonction Ψ appelée potentiel. w_{jk} sont les poids donnés au lien entre le voxel x_j et son voisin x_k . Ces poids peuvent être choisis uniformément égaux à 1, ou bien prendre en compte un facteur de proximité, c'est-à-dire une distance du voxel x_j à x_k . Ils peuvent aussi prendre en compte un facteur de similarité (cf. les poids de Bowsher expliqués dans la partie 1.2.5.7.2.3) ou non.

Nous allons voir plusieurs fonctions potentielles proposées dans la littérature, ainsi que la méthode des poids de Bowsher pour régulariser avec une information anatomique.

1.2.5.7.2.1 Différentes fonctions potentielles

- Le potentiel quadratique a été introduit par [35] : $\Psi : x \in \mathbb{R} \mapsto \frac{x^2}{2}$. Ce potentiel est connu pour lisser énormément les bords entre différentes régions de l'image reconstruite [65],
- le potentiel de Geman [34] : $\Psi : x \in \mathbb{R} \mapsto \frac{x^2}{x^2 + 2\sigma^2}$. Celui-ci pénalise beaucoup moins les différences élevées entre voxels voisins, mais n'est par conséquent pas concave,
- le potentiel de Huber [64] : $\Psi : x \in \mathbb{R} \mapsto \begin{cases} 0, 5x^2, & \text{if } |x| \leq \sigma \\ \frac{|x| - \sigma/2}{\sigma}, & \text{if } |x| > \sigma \end{cases}$. Celui-ci permet des bords plus marqués que la potentiel de quadratique, car il devient linéaire si l'écart entre deux voxels voisins devient plus grand qu'un seuil σ . Le désavantage est de devoir choisir le seuil σ ,
- la pénalité RDP que nous allons expliquer plus en détail dans la partie 1.2.5.7.2.2,
- d'autres fonctions ont été proposées et ne seront pas détaillées ici (cf. [71] pour les retrouver).

1.2.5.7.2.2 Pénalité RDP La pénalité RDP (Relative Difference Penalty) est une pénalité MRF proposée par [65], s'appuyant sur les forces des potentiels présentés ci-dessus. L'objectif de cette méthode est de :

4. La fonction Ψ utilisée par la pénalité RDP dépend aussi de la somme $x_j + x_k$ comme nous allons voir dans la partie 1.2.5.7.2.2.

- construire une pénalité concave (pour assurer l'unicité de la solution),
- favoriser des bords bien définis en pénalisant le moins possible les différences fortes entre voxels voisins. Or, en prenant le cas d'une fonction 1D concave, sa dérivée est forcément croissante. L'intuition dans [65] est alors de chercher une pénalité dont le gradient croît tout en cherchant à se stabiliser quand l'écart entre deux voxels voisins est élevé,
- pénaliser les différences relatives $\frac{|x_j - x_k|}{x_j + x_k}$ plutôt que les différences absolues $|x_j - x_k|$ entre voxels voisins. Comme vu précédemment, les méthodes de Huber [64] et de Geman [34] ont un paramètre σ pour définir quelles différences absolues sont grandes ou non. Plutôt que d'avoir ce paramètre qui dépend énormément de l'image TEP à reconstruire, pénaliser les différences relatives permet de s'adapter à des intensités de voxels variables.

Les auteurs dans [65] ont défini la pénalité de la façon suivante :

$$R(x) = \sum_{j=1}^J \sum_{k \in \mathcal{N}_j} w_{jk} \frac{(x_j - x_k)^2}{(x_j + x_k) + \gamma |x_j - x_k|} \quad (1.56)$$

[65] démontre que la pénalité (1.56) est bien concave⁵, et se comporte de façon proche du MRF avec potentiel quadratique pour des différences relatives inférieures à $\frac{1}{\gamma}$, et de façon linéaire après $\frac{1}{\gamma}$. Ainsi, le paramètre γ définit quelles différences sont petites ou grandes. Un paramètre γ élevé entraîne un comportement linéaire plus rapidement, ce qui autorise des bords plus marqués dans l'image reconstruite.

C'est cette pénalité qu'utilise actuellement GE Healthcare sur ses scanners TEP, couplée à l'algorithme BSREM [2].

1.2.5.7.2.3 Méthode de Bowsher La méthode de Bowsher est un cas particulier de la pénalité MRF (1.55), avec des poids w_{jk} incluant des facteurs de similarité spécifiques à cette méthode. On définit un pourcentage p des voxels x_k appartenant au voisinage \mathcal{N}_j les plus similaires au voxel x_j . Une image supplémentaire (typiquement une image anatomique) est utilisée pour définir la similarité entre deux voxels voisins. Dans la méthode de Bowsher avec utilisation d'image anatomique, la similarité de deux voisins augmente si la différence entre les intensités de ces deux voxels dans l'image anatomique diminue. Les facteurs de similarité sont alors égaux à 1 si le voxel x_k répond au critère de similarité précédent, 0 sinon.

Par ailleurs, dans le calcul théorique de la dérivée de la pénalité R par rapport au voxel courant x_j , deux types de termes sont présents :

- les termes associés aux voxels x_k les plus similaires au voxel courant x_j (pondérés par w_{jk}),
- les termes associés à la similarité du voxel x_j avec les voxels x_k de son voisinage (pondérés par w_{kj}).

La méthode asymétrique [96] [78] se distingue de la version symétrique [29] par ce calcul de la dérivée de la pénalité. La version symétrique prend en compte les deux types de termes alors que la version asymétrique ne garde que les termes associés aux voxels x_k les plus similaires au voxel courant x_j . La convergence de la méthode de Bowsher asymétrique n'a pas été démontrée, mais est vérifiée en pratique. De plus, celle-ci améliore les performances des algorithmes de reconstruction en termes de compromis biais variance par rapport à la méthode symétrique [96] [78].

Dans la suite, en parlant de méthode de Bowsher, ou poids de Bowsher, nous parlerons seulement de la version asymétrique de la méthode de Bowsher.

En plus des méthodes itératives dites classiques, l'évolution des méthodes d'apprentissage profond (deep learning en anglais) a conduit à une nouvelle vague d'algorithmes de reconstruction utilisant des réseaux de neurones. Ils se déclinent en plusieurs catégories, selon leur

5. La pénalité RDP est effectivement concave si les voxels de l'image sont positifs. La pénalité RDP ne peut donc pas être utilisée avec les algorithmes autorisant les valeurs négatives (cf. partie 1.2.5.6).

utilisation ou non d'une base de données (respectivement algorithme supervisé et non supervisé) et selon leur utilisation ou non du modèle physique de la TEP.

1.2.6 Méthodes de reconstruction avec réseau de neurones

1.2.6.1 Méthodes supervisées

1.2.6.1.1 Reconstruction directe avec réseau de neurones Les réseaux de neurones peuvent être entraînés à réaliser une tâche à partir d'une base de données. Par exemple, des réseaux de neurones ont été entraînés pour prendre en entrée un sinogramme, et directement proposer une image reconstruite associée [42]. Ce type de réseau n'utilise pas le modèle direct ni la physique de l'imagerie TEP. Une fois le réseau de neurones entraîné, la reconstruction d'image est beaucoup plus rapide que la reconstruction par méthode itérative. Cependant, les images produites sont très dépendantes de la base de données d'entraînement et peuvent présenter des artefacts. La qualité des images générées par le réseau de neurones peut être améliorée à condition d'avoir des données reconstruites d'entraînement de haute qualité, ainsi qu'une plus grande base de données, ce qui est très difficile dans le contexte médical [74].

1.2.6.1.2 Méthodes de unrolling Les méthodes dites de unrolling proposent un compromis entre modèle physique de l'acquisition TEP et apprentissage à partir de données déjà reconstruites. Une méthode de reconstruction itérative peut être déroulée (unrolled) en remplaçant certaines étapes par un réseau de neurones. Par exemple, plutôt que de choisir une fonction de régularisation (ou sa dérivée [60]) ainsi que le paramètre de pénalité associé, un réseau de neurones apprend à régulariser une image à partir d'une base de données constituées de paires d'images (images reconstruites avec peu de statistiques - images reconstruites avec un haut niveau de statistiques) [60] [39]. Ces méthodes sont de fait plus interprétables et nécessitent aussi moins de données que les méthodes directes pour obtenir de bonnes performances [74], étant donné la présence de la prise en compte de la modélisation physique.

1.2.6.1.3 ADMM Plug and Play La méthode Plug and Play peut aussi être employée pour la reconstruction d'images [95]. Grâce à l'algorithme ADMM, cette méthode décompose un problème de reconstruction PML en deux sous-problèmes : un problème de reconstruction avec une pénalité quadratique simple cherchant à s'ajuster à une image, et un problème de débruitage. Le problème PML initial incorpore une fonction de régularisation, qui est remplacée de façon heuristique par un débruiteur dans l'algorithme Plug and Play. Par conséquent, la deuxième étape de l'ADMM devient une étape de débruitage, où l'on peut utiliser le débruiteur de son choix. Dans [89], un réseau de neurones préalablement entraîné pour cette tâche est utilisé.

1.2.6.2 Méthodes non supervisées

Pour éviter de recourir à une base de données d'images, des méthodes de reconstruction non supervisées, c'est-à-dire ne nécessitant pas de base de données, ont été développées. Celle-ci s'appuient sur le principe du Deep Image Prior (DIP) [92] que nous allons voir dans le chapitre suivant (partie 2.2.2).

Avant d'aborder plus en détail la méthode du Deep Image Prior, terminons ce chapitre avec les applications cliniques étudiées lors de cette thèse.

1.3 Applications cliniques d'intérêt pour cette thèse

1.3.1 Machines TEP utilisées en clinique

Une image anatomique (obtenue par acquisition TDM ou IRM) apporte des informations sur les structures et les tissus dans le corps humain. Elle est complémentaire à l'image TEP qui

apporte seulement une information au niveau du métabolisme du patient et aide au diagnostic. L'image anatomique permet une meilleure localisation des informations vues sur l'image TEP.

Au CHU Hôtel Dieu de Nantes, le service de médecine nucléaire est actuellement doté de deux systèmes multimodalités :

- le Biograph Vision 600, un système TEP/TDM commercialisé par Siemens Healthineers,
- le Biograph mMR, un système TEP/IRM commercialisé par Siemens Healthineers.

Les systèmes TEP actuels sont couplés à une modalité d'imagerie anatomique pour entre autres permettre le recalage et la fusion des deux modalités, ainsi que d'effectuer la correction d'atténuation (cf. partie 1.1.2.4).

Dans ce travail, les méthodes développées cherchent à exploiter des informations anatomiques détaillées pour améliorer la reconstruction des images TEP. L'IRM fournit davantage de détails structurels au niveau des tissus de densités similaires, contrairement au TDM qui ne donne qu'une information de densité. Nous nous focaliserons donc sur les applications TEP/IRM.

Deux applications cliniques sont d'intérêt au CHU de Nantes et réalisées sur le Biograph mMR : l'imagerie [18F]-FDG et la radioembolisation à ^{90}Y .

1.3.2 Applications au [18F]-FDG

Préalablement à un examen TEP, un produit radioactif est injecté dans le corps du patient. Le produit le plus commun est le [18F]-FDG. Nous parlerons plus simplement de FDG par la suite. Cette molécule est composée de deux parties :

- le désoxyglucose, une molécule analogue au glucose. Les cellules tumorales consomment excessivement du glucose, ce qui permet de les cibler pendant un examen TEP FDG.
- un atome de ^{18}F . Cet isotope du fluor est un émetteur β^+ . Il est fixé au désoxyglucose pour permettre de localiser les molécules injectées, et donc les cellules tumorales.

Après injection, le patient doit attendre une heure avant de réaliser l'examen TEP, pour un examen durant entre 20 et 30 minutes.

1.3.3 Applications en radioembolisation à ^{90}Y

1.3.3.1 L'Yttrium-90 (^{90}Y)

L'Yttrium-90 (^{90}Y) est un isotope radioactif se désintégrant principalement sous forme β^- . Cela en fait un candidat pour la radiothérapie interne sélective, c'est-à-dire pour irradier des cellules cancéreuses. Il se désintègre aussi par émission β^+ , mais de façon très peu probable (rapport d'embranchement de $31,86 \times 10^{-6}$ [15]). Une imagerie TEP à partir de ce radioélément est donc possible.

1.3.3.2 Radioembolisation

La radiothérapie interne sélective (Selective Internal Radiation Therapy - SIRT), également appelée radioembolisation, est un traitement pouvant être utilisé pour un cancer primaire hépatique, ou pour un cancer secondaire avec métastases hépatique exclusive. L'objectif est de détruire localement les cellules tumorales du foie grâce à un émetteur β^- , ici ^{90}Y . Le sang alimente le parenchyme hépatique sain à travers la veine porte, contrairement aux masses tumorales qui sont alimentées via l'artère hépatique. Ainsi, pour cibler le plus possible les cellules tumorales tout en préservant les cellules saines, la procédure implique l'injection de microsphères contenant de ^{90}Y à travers l'artère fémorale pour ensuite atteindre l'artère hépatique (cf. figure 1.16). Les microsphères se logent ensuite dans des petits vaisseaux sanguins irriguant le foie. La portée de ces microsphères dans les tissus mous est en moyenne de 2,5 mm et peut atteindre jusqu'à un maximum de 11 mm [55]. Pendant cette thérapie, l'objectif est de garantir que le rayonnement atteigne uniquement les cellules tumorales, minimisant ainsi les dommages aux tissus sains.

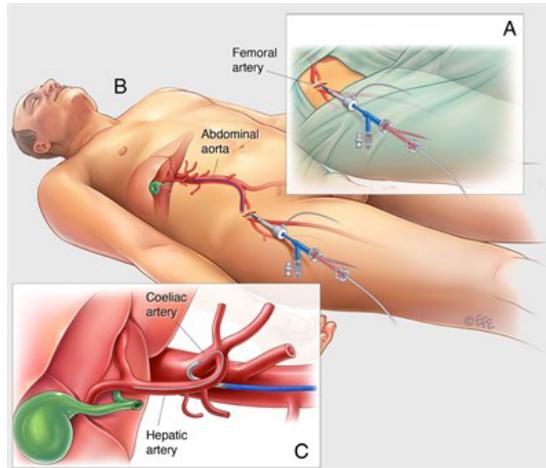


FIGURE 1.16 – Injection pour un traitement par radioembolisation (source : www.sirtex.com).

Après l'injection, plusieurs imageries à différents jours d'intervalle sont réalisées par Tomographie par Émission MonoPhotonique (TEMP) ou par TEP, utilisant la très faible émission de positons issue des microsphères d' ^{90}Y . Le rapport d'embranchement de ^{90}Y étant extrêmement faible (cf. partie 1.3.3.1), le nombre de coïncidences détectées lors d'une acquisition TEP réalisée après une thérapie par radioembolisation à ^{90}Y est très faible. De plus, pour un système TEP utilisant des matériaux LSO ou LYSO, la fraction de coïncidences fortuites venant des détecteurs est extrêmement importante (cf. partie 1.2.1.2.3). Reconstruire une image dans ce cadre est un challenge étant donné la faible quantité de signal utile parmi un fort bruit de fond.

Comme expliqué dans la partie 1.2.5.5.4, les images reconstruites présentent un biais positif dans les régions froides pour ce type d'applications à faible statistiques d'acquisition lors de l'utilisation d'algorithmes de reconstruction avec contrainte de non-négativité dans l'image. Il est donc nécessaire d'utiliser des algorithmes de reconstruction capable de réduire ce biais. Dans le chapitre suivant, nous allons présenter l'algorithme de reconstruction DNA développé lors de cette thèse.

Chapitre 2

Développement de l’algorithme DNA : reconstruction d’images TEP utilisant du Deep Image Prior et une contrainte de non-négativité dans l’espace de projection

2.1 Introduction

La TEP est utilisée pour faire une image de la distribution de la radioactivité après l’injection d’un médicament radiopharmaceutique. L’étape de reconstruction de l’image TEP consiste à trouver une représentation fidèle de la distribution radioactive à partir des données bruitées mesurées par le système d’imagerie. Il s’agit d’un problème inverse mal posé, généralement résolu par l’optimisation de fonctions objectives à l’aide d’algorithmes itératifs [72] (cf. partie 1.2.5). Pour traiter le problème de l’amplification du bruit entre les données et l’image, les méthodes de reconstruction actuellement utilisées en clinique s’appuient soit sur :

- des algorithmes ML, accélérés par sous-ensembles, avec arrêt précoce et post filtrage (*e.g.* OSEM [52], cf. partie 1.2.5.5.2),
- des algorithmes PML convergents où les variations importantes des valeurs des voxels voisins sont pénalisées (*e.g.* BSREM [1] + RDP [65], cf. parties 1.2.5.5.3 et 1.2.5.7.2.2).

Une autre catégorie d’algorithmes de reconstruction a été développée au cours de la dernière décennie à l’aide de méthodes d’apprentissage profond (cf. partie 1.2.6). Tout d’abord, l’apprentissage profond classique avec des réseaux de neurones entraînés sur des bases de données a été utilisé pour la reconstruction TEP [74]. Cependant, ce type d’algorithme nécessite la construction d’une base de données, dont la taille est souvent limitée pour l’imagerie TEP. De plus, ces méthodes peuvent entraîner des artefacts dans l’image reconstruite [42]. Plus récemment, le concept du Deep Image Prior (DIP) [92] a été introduit comme un moyen non supervisé de débruiter ou de reconstruire une image, améliorant ainsi la qualité de l’image. Il a été appliqué à des acquisitions TEP statiques :

- comme une étape de post traitement après la reconstruction itérative [24; 25; 58; 67; 68],
- intégrée dans la reconstruction [17; 37; 69],
- en tant que méthode de reconstruction de bout en bout (end-to-end reconstruction) [46; 47; 53; 81],

Le DIP a aussi été appliqué pour le débruitage [43; 88] et la reconstruction d’images TEP dynamiques [38; 107].

Premièrement, l'utilisation du DIP comme débruiteur sur les images TEP reconstruites a été étudié et a donné de meilleurs résultats que les méthodes de débruitage de l'état de l'art telles que le filtrage gaussien [25]. De plus, l'utilisation d'une image anatomique en entrée du DIP a amélioré la qualité de l'image TEP révélant davantage les structures cérébrales [25]. L'algorithme DIPRecon [37] a ensuite été développé, et cherche à optimiser la vraisemblance avec la contrainte que l'image reconstruite soit la sortie d'un réseau de neurones DIP. Celui-ci dérive un algorithme ADMM pour découpler l'entraînement du DIP de l'optimisation de la vraisemblance de Poisson. Par rapport au simple débruitage d'une image reconstruite, la méthode DIPRecon a permis d'améliorer le contraste des images reconstruites. Comme dans [25], l'utilisation d'une image anatomique en entrée du DIP a également permis d'améliorer les détails des structures cérébrales.

Par ailleurs, les algorithmes de reconstruction itératifs ayant une contrainte de non-négativité dans l'espace image [82] sont connus pour provoquer un biais positif dans les régions de faible activité (cf. partie 1.2.5.5.4), en particulier dans les acquisitions à faible statistiques [16]. Or, le DIPRecon résout en interne un problème de reconstruction avec contrainte de non-négativité, et réalise un débruitage DIP avec une architecture imposant des valeurs de sortie non négatives.

En outre, comme vu dans la partie 1.2.5.6, plusieurs algorithmes de reconstruction ont été développés pour autoriser les valeurs négatives dans l'image et ainsi réaliser une réduction de biais. Dans [62], une comparaison a été effectuée entre plusieurs algorithmes de reconstruction proposant une réduction de biais via la suppression de la contrainte de non-négativité, dans le cas de la radioembolisation à ^{90}Y . Les auteurs ont montré que l'algorithme empirique APPGML (cf. partie 1.2.5.6.3) permet d'obtenir la meilleure réduction de biais dans les régions froides ainsi que les meilleurs compromis entre bruit et biais. En parallèle, l'algorithme ADMM-Reg [59] a été développé en assouplissant la contrainte de non-négativité dans l'espace image en ne l'appliquant que dans l'espace des projections, ce qui préserve mathématiquement l'hypothèse de la loi de Poisson.

Dans ce chapitre, nous allons présenter l'algorithme de reconstruction DNA (DIP Nested ADMM) développé lors de cette thèse. Le DNA a été conçu comme une amélioration du DIPRecon, supprimant ses biais intrinsèques en autorisant des voxels négatifs. La fonction d'activation finale ReLU (Rectified Linear Unit) du réseau DIP a été supprimée et l'algorithme ADMM-Reg [59] a été utilisé. Nous avons dérivé le DNA en utilisant l'algorithme ADMM-Reg pour la réduction de biais qu'il propose ainsi que pour sa dérivation mathématique. Ce chapitre reprend les explications et résultats d'un article soumis récemment dans le journal IEEE TMI (cf. partie A.1). Le texte soumis est disponible à la fin du manuscrit.

Voyons tout d'abord quelques définitions autour de l'entraînement et de l'architecture d'un réseau de neurones, avant de présenter le concept du DIP et comment débruiter une image de façon non supervisée.

2.2 Contexte

2.2.1 Réseau de neurones

2.2.1.1 Entraînement d'un réseau de neurones

Dans ce paragraphe, nous ne définirons que les termes liés à un réseau de neurones (architecture, entraînement) et qui seront utiles pour cette thèse. L'objectif ici est d'introduire des termes spécifiques à un réseau de neurones, sans plonger dans les détails d'un entraînement supervisé.

Le but d'un réseau de neurones est de répondre à une tâche (débruitage d'une image dans notre cas). Pour cela, un réseau de neurones est défini mathématiquement comme une fonction non linéaire paramétrée par des poids. Il prend une entrée, la traite et calcule une sortie. Dans notre cas, on veut débruiter une image. Une fonction non linéaire est utilisée pour relier une image d'entrée à une image de sortie qui sera l'image débruitée. On appelle entraînement du réseau de neurones l'étape pour déterminer les poids associés à une architecture donnée, et donc

une fonction non linéaire capable de réaliser la tâche souhaitée. Pour entraîner un réseau de neurones, l'idée est de comparer sa sortie avec une image étiquette (label image en anglais). Cette image est fournie par l'utilisateur pour aider le réseau de neurones à savoir quel type d'image produire. Cette comparaison s'effectue mathématiquement grâce à une fonction de coût. L'idée est alors de minimiser cette fonction de coût en modifiant les poids du réseau pour faire en sorte que celui-ci soit capable de produire une image proche de l'image étiquette.

Voici l'exemple de l'entraînement d'un réseau de neurones avec une fonction de coût étant la MSE (Mean Square Error) :

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \|x_{label}^i - f(\theta|x_{input}^i)\|_2^2 \quad (2.1)$$

avec

- $\{x_{label}^i\}_{i=1}^N$: N images étiquettes que le réseau essaye de produire,
- $\{x_{input}^i\}_{i=1}^N$: N images d'entrées que le réseau prend en entrée,
- f : architecture du réseau de neurones,
- θ : poids du réseau.

Pendant son entraînement, un réseau de neurones supervisé utilise un nombre N de paires d'images, qui peut être de plus en plus grand en fonction notamment du nombre de paramètres θ à apprendre.

Un algorithme itératif est utilisé pour résoudre ce problème d'optimisation, appelé optimiseur, qui s'appuie en général sur l'algorithme de descente de gradient. Le pas de descente de gradient est appelé taux d'apprentissage (taux d'apprentissage en anglais).

2.2.1.2 Architecture d'un réseau de neurones

Nous allons voir dans cette partie certaines couches et fonctions utilisées à l'intérieur des réseaux de neurones lors de cette thèse. Certains termes seront réutilisés en anglais par la suite :

- *Couche convolutionnelle* : une ou plusieurs images d'entrée sont convoluées par un ou plusieurs noyaux de convolutions (dont les paramètres font partie des poids θ) et produit une ou plusieurs images. Le nombre d'images en sortie de cette couche est appelée nombre de canaux. Un cas particulier est la couche convolutionnelle avec pas de 2 (stride 2 en anglais) : on ne considère qu'un voxel sur 2 dans les images en entrée de cette couche pour effectuer la convolution, permettant de sous-échantillonner spatialement une image.
- *Interpolation bilinéaire* : une interpolation bilinéaire est effectuée sur une image d'entrée pour doubler son nombre de voxels selon toutes les dimensions. Le nombre de couches d'interpolation bilinéaire est appelée profondeur du réseau convolutionnel,
- *Normalisation par lots - Batch normalization en anglais* : une image en entrée de cette couche est standardisée par une moyenne et une variance (faisant partie des poids θ associés à cette couche),
- *Sauts de connexion - Sauts de Connexion (SC) en anglais* : connexions permettant de passer de l'information entre deux endroits du réseau de neurones en sautant certaines couches,
- *Fonction d'activation* : un réseau de neurones est une succession d'opérations linéaires entre une entrée et les poids θ et d'une fonction d'activation ayant pour but d'introduire de la non linéarité. En voici plusieurs exemples :
 - ReLU - Rectified Linear Unit : $x \mapsto \max(0, x)$
 - LeakyReLU : $x \mapsto \begin{cases} x, & x \geq 0 \\ \alpha_L x, & x < 0 \end{cases}$, avec α_L un hyperparamètre défini par l'utilisateur en amont de l'entraînement contrôlant la pente de la droite pour les valeurs de x négatives,
 - linéaire : $x \mapsto x$.

Les termes ci-dessus vont permettre maintenant d'introduire le concept du Deep Image Prior.

2.2.2 Deep Image Prior (DIP)

Le Deep Image Prior (DIP) a été développé par [92] comme un moyen non supervisé de résoudre des problèmes inverses, originellement avec un réseau de neurones de type U-Net [75]. Ses performances ont d'abord été étudiées pour le débruitage, la super résolution et l'inpainting [92]. Le concept a ensuite été étudié pour la résolution de problèmes inverses plus généraux comme celui de la TEP, ce qui a donné des algorithmes de reconstruction de bout en bout [46; 47; 53; 81]. La tâche de débruitage avec le DIP s'écrit comme le problème d'optimisation suivant :

$$\theta^* = \arg \min_{\theta} \|x_{label} - f(\theta|z)\|_2^2, \quad x^* = f(\theta^*|z) \quad (2.2)$$

avec x_{label} l'image bruitée que l'on cherche à débruiter, x^* l'image débruitée estimée après entraînement du DIP, f l'architecture du réseau de neurones, θ les poids du réseau de neurones appris pendant l'entraînement, z l'entrée du réseau.

Ce problème d'optimisation s'apparente à l'entraînement d'un réseau de neurones supervisé, mais avec une seule paire d'images dans la base de données (entrée z , étiquette x_{label}). Un réseau de neurones supervisé aurait besoin de labels étiquetés par l'homme avant l'entraînement. Le DIP est précisément non supervisé car cette image est l'image que l'on cherche à débruiter. L'image d'entrée z peut être choisi de façon aléatoire [92] ou utilisant une information anatomique (provenant d'une acquisition TDM ou IRM).

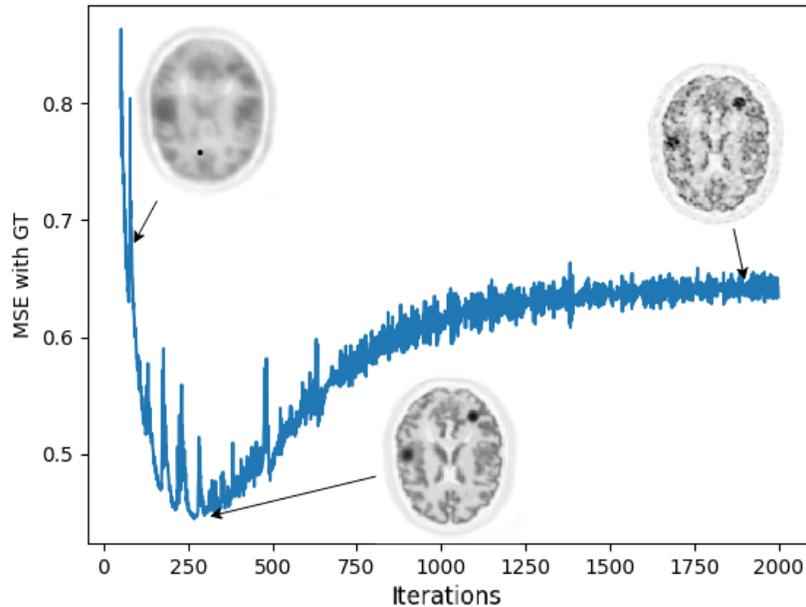


FIGURE 2.1 – Phénomène d'apprentissage précoce puis surapprentissage du DIP.

Pour débruiter une image, la sortie du DIP tente de s'ajuster à l'image bruitée comme indiqué dans (2.2). Au cours de l'optimisation, l'image estimée se rapproche d'abord d'une version débruitée de l'image bruitée, qui est l'image souhaitée, avant d'ajouter les hautes fréquences indésirables de l'image bruitée [19; 83]. On dit alors que le DIP entre dans son régime de surapprentissage : le fait d'effectuer des itérations supplémentaires ne contribue plus au débruitage de l'image étiquette. La figure 2.1 illustre cela, avec la variation de la MSE entre la sortie du DIP et la vérité terrain (Ground Truth - GT). La MSE diminue au cours des premières itérations du DIP, puis atteint un minimum global, et augmente quand le bruit est ajouté dans la sortie du DIP [19; 83]. Les auteurs du DIP [92] expliquent que c'est l'architecture du réseau qui agit comme un prior pour produire des images "naturelles" [92]. Le comportement général du DIP de débruitage puis de surapprentissage est appelé l'ELTO (Early Learning Then Overfitting).

Une autre approche pour comprendre le principe du DIP est de voir le réseau de neurones seulement comme une fonction non linéaire. Le but est de trouver les paramètres d'une fonction permettant de passer d'une image d'entrée à une image TEP, en résolvant le problème d'optimisation (2.2). Finalement, le seul rapprochement avec l'apprentissage profond est l'utilisation d'un réseau de neurones pour représenter cette fonction non linéaire.

Voyons maintenant l'algorithme DIPRecon qui a utilisé en premier le DIP pour la reconstruction TEP.

2.2.3 Algorithme DIPRecon

L'algorithme DIPRecon résout le problème d'optimisation sous contrainte suivant [37] :

$$\begin{cases} \min_x -L(y|x), \\ \text{subject to } x = f(\theta|z). \end{cases} \quad (2.3)$$

Les auteurs dans [37] ont dérivé l'algorithme DIPRecon à partir de l'algorithme ADMM, ce qui donne le système d'équations suivant :

$$x^{n+1} = \arg \max_x L(y|x) - \frac{\rho}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2, \quad (2.4a)$$

$$\theta^{n+1} = \arg \min_{\theta} \|x^{n+1} - f(\theta|z) + \mu^n\|_2^2, \quad (2.4b)$$

$$\mu^{n+1} = \mu^n + x^{n+1} - f(\theta^{n+1}|z), \quad (2.4c)$$

où ρ est le paramètre de pénalité d'ADMM, et μ le multiplicateur de Lagrange.

L'équation (2.4a) est résolue par une méthode de transfert d'optimisation, en choisissant la fonction de substitution de la vraisemblance proposée par [26] :

$$P(x | x^n) = \sum_{j=1}^J P(x_j | x^n), \quad (2.5)$$

avec

$$P(x_j | x^n) = A_{.j} \left(\hat{x}_{j,\text{EM}}^{n+1} \ln x_j - x_j \right) - \frac{\rho}{2} [x_j - f(\theta | z)_j^n + \mu_j^n]^2, \quad (2.6)$$

$$\text{avec } A_{.j} = \sum_{i=1}^I A_{ij} \text{ et } \hat{x}_{j,\text{EM}}^{n+1} = \frac{x_j^n}{\sum_{i=1}^I A_{ij}} \sum_{i=1}^I A_{ij} \frac{y_i}{\sum_{k=1}^J A_{ik} x_k + \bar{b}_i}.$$

La fonction P ainsi définie est séparable en les voxels x_j , ce qui rend sa maximisation possible voxel par voxel. Le calcul de la dérivée de la fonction de substitution donne après regroupement des termes :

$$\frac{\partial P}{\partial x_j} = -\rho x_j^2 + [\rho(f(\theta^n | z) - \mu_j^n) - A_{.j}] x_j + A_{.j} \hat{x}_{j,\text{EM}}^{n+1}. \quad (2.7)$$

On cherche donc à annuler sa dérivée par rapport à chaque voxel, ce qui revient à trouver les solutions d'un trinôme du second degré en x_j . On pose :

$$b = f(\theta^n | z)_j - \mu_j^n - \frac{A_{.j}}{\rho}, \quad (2.8)$$

et

$$d = \sqrt{b^2 + 4\rho\hat{x}_{j,\text{EM}}^{n+1}A_{.j}}, \quad (2.9)$$

$$= \sqrt{\left[f(\theta^n | z)_j - \mu_j^n - A_{.j}/\rho\right]^2 + 4\rho\hat{x}_{j,\text{EM}}^{n+1}A_{.j}}. \quad (2.10)$$

Cette racine est bien définie car le discriminant de ce trinôme égal à $b^2 + 4\rho\hat{x}_{j,\text{EM}}^{n+1}A_{.j}$ est bien positif. Les solutions du trinôme du second degré sont alors données par :

$$\begin{cases} x_j^{n+1} = \frac{b+d}{2} \\ x_j^{n+1} = \frac{b-d}{2} \end{cases} \quad (2.11)$$

Or les voxels de l'image x_j^{n+1} doivent être positifs (à cause notamment du logarithme dans l'expression (2.6)). De plus, $d > b$ donc la solution $x_j^{n+1} = \frac{b+d}{2}$ est toujours positive et supérieure à la solution $x_j^{n+1} = \frac{b-d}{2}$.

L'expression finale des itérés de l'étape (2.4a) est donc :

$$x_j^{n+1} = \frac{b+d}{2} \quad (2.12)$$

$$= \frac{1}{2} \left(f(\theta^n | z)_j - \mu_j^n - \frac{A_{.j}}{\rho} \right) + \frac{1}{2} \sqrt{\left[f(\theta^n | z)_j - \mu_j^n - A_{.j}/\rho\right]^2 + 4\rho\hat{x}_{j,\text{EM}}^{n+1}A_{.j}} \quad (2.13)$$

La résolution de l'équation (2.4a) induit donc une contrainte de non négativité sur l'image x^{n+1} . Quant à l'équation (2.4b), elle correspond à l'optimisation du DIP. Dans [37], la fonction d'activation finale est une ReLU qui produit une image non négative $f(\theta^{n+1}|z)$. Cependant, l'image μ^{n+1} peut contenir des voxels négatifs, ce qui implique que l'ensemble des images reconstruites possibles $f(\theta^{n+1}|z)$ dans (2.4b) est plus petit que l'ensemble global des solutions sans contrainte de non-négativité. Le même raisonnement est aussi valable pour (2.4a) en considérant l'image cible $\mu^n - f(\theta^n|z)$ dans le terme de régularisation quadratique.

Dans la suite, nous allons voir comment est dérivée l'algorithme DNA, s'appuyant sur les algorithmes DIPRecon et ADMM-Reg (cf. partie 1.2.5.6.4) de la littérature.

2.3 Développement de l'algorithme DNA

2.3.1 Dérivation mathématique

L'algorithme DNA a été conçu comme une amélioration de l'algorithme DIPRecon, autorisant les valeurs négatives en supprimant la ReLU finale du réseau du DIP et en transférant la contrainte de non-négativité de l'espace image dans l'espace de projection. L'idée pour développer le DIP Nested ADMM (DNA) est de combiner la contrainte de l'algorithme DIPRecon [37] et une contrainte bivariée inspirée de celle de l'algorithme ADMM-Reg [59] sous la forme du problème d'optimisation suivant :

$$\begin{cases} \min_x -L(y|x) + g(\bar{y}, y) \\ \text{subject to } x = f(\theta|z), \end{cases} \quad (2.14)$$

avec

$$g(\bar{y}, y) = \sum_i g_i(\bar{y}_i, y_i), \quad g_i(\bar{y}_i, y_i) = \begin{cases} \infty, & \bar{y}_i < 0 \\ \infty, & \bar{y}_i = 0, y_i > 0 \\ 0, & \text{sinon} \end{cases}$$

Nous avons modifié la fonction barrière g^{Lim} proposée dans [59] (cf. partie 1.2.5.6.4) par la fonction barrière g de sorte que \bar{y}_i soit autorisé à être égal à zéro si et seulement si y_i est nul (cf. annexe B.3 pour les détails mathématiques).

Ce nouveau problème d'optimisation est le même que celui que résout l'algorithme DIPRecon dans [37], avec la fonction coût qui est la somme de la vraisemblance et de la fonction barrière g . On peut le résoudre alors par la méthode ADMM, ce qui donne les mêmes itérés que pour DIPRecon (cf. équations (2.4)), avec le terme $g(\bar{y}, y)$ en plus de la vraisemblance dans l'équation (2.15a) :

$$\begin{aligned} x^{n+1} &= \arg \min_x -L(y|x) + g(\bar{y}, y) \\ &\quad + \frac{\rho_1}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2, \end{aligned} \quad (2.15a)$$

$$\theta^{n+1} = \arg \min_{\theta} \|x^{n+1} - f(\theta|z) + \mu^n\|_2^2 \quad (2.15b)$$

$$\mu^{n+1} = \mu^n + x^{n+1} - f(\theta^{n+1}|z), \quad (2.15c)$$

ρ_1 est le paramètre de pénalité ADMM relié à la contrainte du DIP, et μ est le multiplicateur de Lagrange associé au problème d'optimisation (2.14).

L'équation (2.15a) est résolue en dérivant un deuxième algorithme ADMM. Les équations ADMM sont les mêmes que celles proposées par [59], la fonction barrière g^{Lim} étant remplacée par g . Le terme R dans (1.52) est ici une fonction de régularisation quadratique simple avec le paramètre de pénalité ρ_1 de (2.15a). Un ADMM imbriqué est donc dérivé de la même façon que les équations (1.51) d'ADMM-Reg pour résoudre (2.15a) dans le cadre du premier ADMM (équations (2.15)), ce qui donne un ensemble de cinq équations :

$$\begin{aligned} x^{n+1,k+1} &= \arg \min_x h_x(x), \\ \text{avec } h_x(x) &= \frac{\rho_1}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2 + \frac{\rho_2}{2} \|Ax - v^{n+1,k} + u^{n+1,k}\|_2^2, \end{aligned} \quad (2.16a)$$

$$\begin{aligned} v^{n+1,k+1} &= \arg \min_v h_v(v), \\ \text{avec } h_v(v) &= 1^T(v + \bar{b}) - y^T \ln(v + \bar{b}) + g(v + \bar{b}, y) + \frac{\rho_2}{2} \|Ax^{n+1,k+1} - v + u^{n+1,k}\|_2^2, \end{aligned} \quad (2.16b)$$

$$u^{n+1,k+1} = u^{n+1,k} + Ax^{n+1,k+1} - v^{n+1,k+1}, \quad (2.16c)$$

$$\theta^{n+1} = \arg \min_{\theta} \|x^{n+1} - f(\theta|z) + \mu^n\|_2^2, \quad (2.16d)$$

$$\mu^{n+1} = \mu^n + x^{n+1} - f(\theta^{n+1}|z). \quad (2.16e)$$

où ρ_2 est le paramètre de pénalité ADMM et u le multiplicateur de Lagrange dérivés tel que ADMM-Reg (cf. partie 1.2.5.6.4).

L'initialisation et la résolution des sous-problèmes d'optimisation (2.16a) (2.16b) (2.16d) sont présentées dans la prochaine partie.

2.3.2 Méthodes pour résoudre chaque équation de l'algorithme DNA

Nous représentons l'ensemble des équations du DNA dans le diagramme 2.2. Chaque bloc du diagramme 2.2 est itératif. Nous définissons les termes suivants :

- le nombre de sous-itérations internes est le nombre d'itérations choisi en pratique pour résoudre l'équation (2.16a) (bloc vert du diagramme 2.2),
- le nombre d'itérations internes (K) est le nombre d'itérations choisi en pratique à l'intérieur du bloc bleu du diagramme 2.2. L'itération interne courante est notée k ,

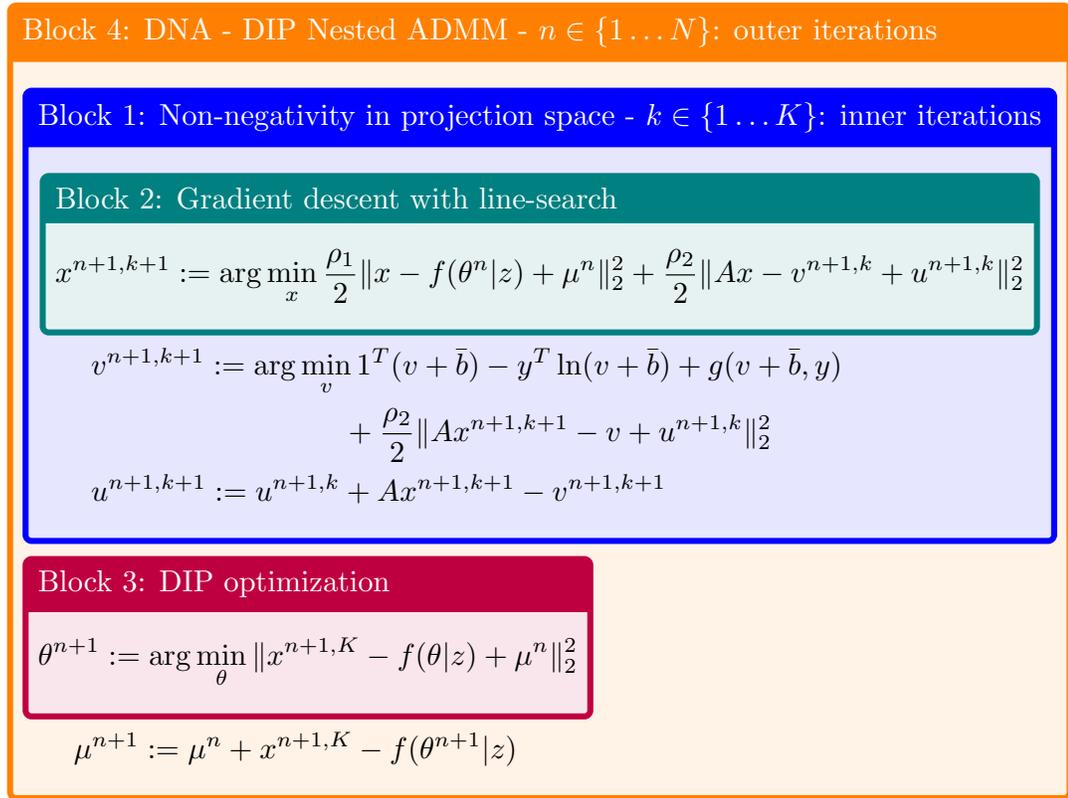


FIGURE 2.2 – Ensemble des équations de l'algorithme DNA. Chaque bloc de ce diagramme est itératif (cf. partie 2.3.2).

- le nombre d'itérations du DIP est le nombre d'itérations choisi en pratique pour résoudre l'équation (2.16d) (bloc rouge du diagramme 2.2),
- le nombre d'itérations externes (N) est le nombre d'itérations de l'algorithme DNA (bloc orange du diagramme 2.2). L'itération externe courante est notée n .

2.3.2.1 Mise à jour de x

L'équation (2.16a) est résolue par un algorithme de descente de gradient. Les auteurs dans [12] expliquent qu'en pratique, les sous problèmes d'optimisation dérivés d'un ADMM peuvent être résolus sans aller à convergence, sous certaines conditions (cf. partie 3.4.4 dans [12]). En s'inspirant de cette remarque, nous avons décidé de fixer le nombre de sous-itérations internes à 1 et d'initialiser la descente de gradient par :

- la sortie DIP calculée à l'itération externe précédente : $x^{n+1,0} = f(\theta^n|z)$ pour la première itération interne,
- l'image $x^{n+1,k}$ calculée à l'itération interne précédente sinon.

On obtient alors la mise à jour suivante :

$$\begin{cases} x^{n+1,0} = f(\theta^n|z), \\ x^{n+1,k+1} = x^{n+1,k} - \zeta^{n+1,k} \nabla_x h_x(x^{n+1,k}), \end{cases} \quad (2.17)$$

avec le gradient :

$$\begin{aligned} \nabla_x h_x(x^{n+1,k}) &= \rho_2 A^T [Ax^{n+1,k} - (v^{n+1,k} - u^{n+1,k})] \\ &\quad + \rho_1 [x^{n+1,k} - (f(\theta^n|z) - \mu^n)]. \end{aligned} \quad (2.18)$$

Enfin, une recherche linéaire est réalisée pour choisir le pas $\zeta^{n+1,k}$ proposant la plus grande descente vers le minimum de (2.16a). Sa valeur est donnée par l'équation suivante¹ :

$$\zeta^{n+1,k} = \frac{\|\nabla_x h_x(x^{n+1,k})\|^2}{\rho_2 \|A \nabla_x h_x(x^{n+1,k})\|^2 + \rho_1 \|\nabla_x h_x(x^{n+1,k})\|^2}. \quad (2.19)$$

2.3.2.2 Mise à jour de v

L'optimisation (2.16b) est exactement la même que dans l'équation de mise à jour de v de l'algorithme ADMM-Reg (1.51b). Cette étape se résout de façon analytique en annulant la dérivée de h_v et en choisissant la solution du problème quadratique la plus stable numériquement, ce qui donne :

$$v_i^{n+1,k+1} = \begin{cases} [Ax^{n+1,k+1}]_i + u_i^{n+1,k} - \frac{1}{\rho_2}, & y_i = 0 \\ \sqrt{\kappa^2 + \gamma} - \kappa, & y_i > 0, \kappa < 0 \\ \frac{\gamma}{\sqrt{\kappa^2 + \gamma} + \kappa}, & y_i > 0, \kappa \geq 0 \end{cases} \quad (2.20)$$

où :

$$\kappa = 0,5 \left(\frac{1}{\rho_2} + \bar{b}_i - u_i^{n+1,k} - [Ax^{n+1,k+1}]_i \right), \quad (2.21)$$

$$\gamma = \bar{b}_i \left(u_i^{n+1,k} + [Ax^{n+1,k+1}]_i \right) - \frac{\bar{b}_i - y_i}{\rho_2}. \quad (2.22)$$

1. Une recherche linéaire est un calcul mathématique pour déterminer la valeur du pas de descente dans un algorithme de descente de gradient. Les calculs effectués pour déterminer le pas $\zeta^{n+1,k}$ sont proposés en annexe B.2.

Enfin, une étape supplémentaire est réalisée si $y_i = 0$ pour assurer que la contrainte $v_i^{n+1,k+1} + \bar{b}_i \geq 0$ imposée par la fonction barrière g est vérifiée :

$$v_i^{n+1,k+1} = [v_i^{n+1,k+1} + \bar{b}_i]_+ - \bar{b}_i. \quad (2.23)$$

2.3.2.3 Mise à jour automatique du paramètre de pénalité d'ADMM ρ_2

Pour sélectionner automatiquement le paramètre de pénalité d'ADMM ρ_2 , nous avons utilisé la méthode de l'équilibre des résidus relatifs [104], qui a la même politique de mise à jour que la méthode utilisée dans [59], mais qui est invariante à la mise à l'échelle (cf. partie 3.4.1).

Remarque : nous avons choisi de ne pas indexer ρ_2 sur les itérations externes et internes pour simplifier les notations.

2.3.2.4 Débruitage DIP à l'intérieur de la reconstruction

Le problème d'optimisation (2.16d) est une tâche de débruitage avec le DIP de l'image $x^{n+1} + \mu^n$.

Tout comme l'algorithme DIPRecon, le DNA alterne une étape de reconstruction (bloc bleu du diagramme 2.2) et une étape de débruitage par le DIP (bloc rouge du diagramme 2.2). Une dernière étape met à jour le multiplicateur de Lagrange μ pour relier ces deux étapes.

Pour terminer cette partie sur les méthodes utilisées, nous allons décrire l'architecture du réseau de neurones utilisée pour le DIP ainsi que les initialisations choisies pour les algorithmes DNA et DIPRecon.

2.3.3 Architecture du réseau de neurones utilisée pour le DIP

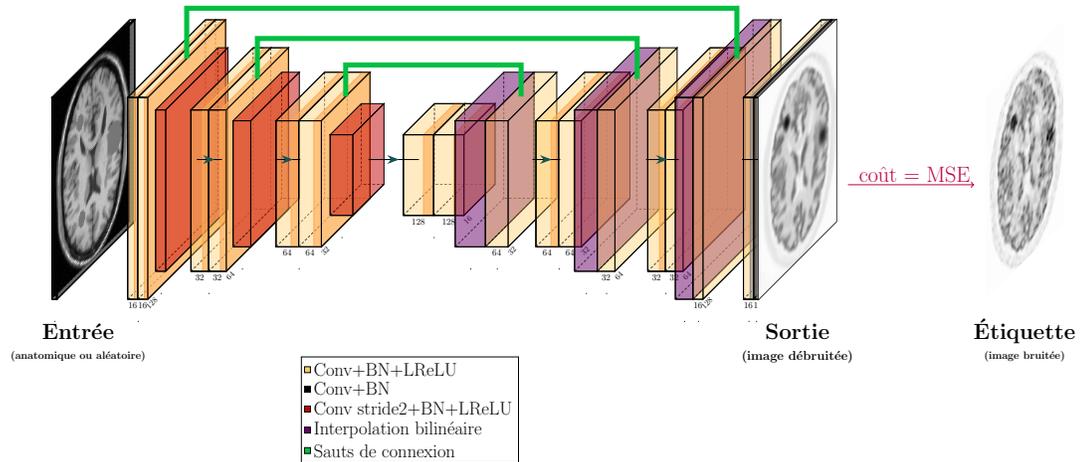


FIGURE 2.3 – Architecture du DIP utilisée pour le DNA. Légende : Conv = couche convolutive ; BN = Batch Normalization ; LReLU = LeakyReLU ; Conv stride2 = couche convolutive avec stride 2.

Le DIP s'appuie initialement sur le fait qu'un réseau de neurones convolutif possède de façon intrinsèque la capacité à générer des images naturelles [92]. L'architecture initialement utilisée s'appuie sur des auto-encodeurs de type U-Net [75]. Ce type d'architecture prend une image en entrée, qui passe à l'intérieur d'un encodeur. Celui-ci encode ce qu'on appelle des caractéristiques de l'image d'entrée. Un décodeur est ensuite associé pour décoder les caractéristiques et reproduire une image. Celle-ci est comparée à une image étiquette (l'image bruitée dans le cas du DIP) ce qui permet de réaliser la tâche demandée (ici un débruitage).

Les auteurs du DIPRecon [37] ont proposée une architecture avec de légères modifications par rapport au U-Net original :

- pour l’encodage, les opérations de sous-échantillonnage sont réalisées par des convolutions avec stride 2, plutôt que des max pooling,
- utiliser des sauts de connexion sous forme d’addition plutôt que sous forme de concaténation,
- pour le décodage, les opérations de sur-échantillonnage sont réalisées par des couches d’interpolation bilinéaire, plutôt que par des convolutions transposées,
- utiliser des fonctions d’activation leakyReLU à l’intérieur du réseau de neurones, et une fonction ReLU à la toute fin plutôt que des fonctions ReLU.

Ces modifications ont pour but d’améliorer la capacité du U-Net à représenter une image naturelle, en supprimant de possibles artefacts du damier, tout en diminuant le nombre de paramètres à apprendre pour faciliter l’entraînement. Pour le DNA, l’architecture du réseau 3D a été choisie comme le même U-Net modifié que celui utilisé dans [37], excepté la ReLU finale qui est retirée comme expliqué en introduction 2.1. L’architecture du réseau 2D est la même que celle 3D, avec des couches convolutionnelles 2D et des couches de normalisation par lots 2D. Cette architecture est montrée dans la figure 2.3.

Initialement, les auteurs dans [92] ont proposé de mettre du bruit uniforme en entrée du DIP. Comme expliqué dans la partie 1.3.1, une image anatomique peut être réalisée par IRM ou TDM en plus de l’acquisition TEP. Pour l’algorithme DNA, nous avons décidé d’utiliser une image anatomique comme entrée du DIP. Cela permet de bénéficier de détails structurels présents dans l’image anatomique [25] [37].

2.3.4 Initialisation des algorithmes DNA et DIPRecon

Les problèmes d’optimisation du DIPRecon et du DNA sont non convexes à cause de la contrainte $x = f(\theta|z)$, et dépendent donc de leur initialisation. Les auteurs dans [37] ont décidé d’initialiser le DIPRecon à partir d’une image pré-reconstruite pour 60 itérations avec l’algorithme MLEM. Cette image est ensuite débruitée par un réseau de neurones DIP pendant 300 itérations avec l’optimiseur L-BFGS. Dans ce chapitre, nous avons choisi une autre initialisation en accord avec les résultats du chapitre 3 (cf. partie 3.3.1 et partie 3.5.1). Une image a d’abord été calculée avec l’algorithme BSREM [1] avec la pénalité RDP [65] en utilisant un paramètre de pénalité assez faible pour favoriser un meilleur contraste au détriment d’un léger bruit. Nous appellerons cette image : image BSREM de départ. Le réseau de neurones est ensuite optimisé avec l’optimiseur Adam pour débruiter cette image BSREM de départ. Pour arrêter l’optimisation du DIP avant d’obtenir l’image BSREM bruitée, nous avons utilisé l’algorithme EMV (Exponential Moving Variance) introduit dans [99] (cf. détails dans la partie 3.3.2.2). Nous appelons cette image : initialisation DIP-BSREM. Les poids du réseau optimisés pendant l’initialisation sont ensuite utilisés pour le prochain débruitage du DIP, et l’initialisation DIP-BSREM a été utilisée comme première valeur d’image $f(\theta^0|z)$ dans (2.16a) et dans (2.4a).

La prochaine partie introduit les expériences réalisées pour étudier les performances du DNA.

2.4 Expériences

Trois expériences ont été réalisées. La première est une expérience de base conçue pour mettre en évidence les artefacts qui peuvent être causés par la ReLU comme fonction d’activation finale lors de l’utilisation du DIP. La deuxième expérience s’appuie sur des simulations analytiques TEP 2D [87] d’un scanner cérébral TEP/IRM et est utilisée pour évaluer la performance de l’algorithme proposé comparé à d’autres méthodes de la littérature. La troisième expérience est réalisée sur des données cérébrales TEP/IRM réelles afin de confirmer l’efficacité de la méthode proposée.

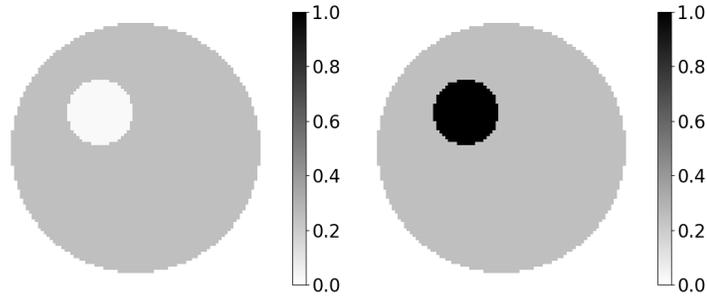


FIGURE 2.4 – Fantôme conçu pour mettre en évidence les artefacts causés par la ReLU (*gauche* : GT TEP avec insert froid ; *droite* : GT TEP avec insert chaud).

2.4.1 Expérience basique ”DIP ReLU”

Pour mettre en évidence les artefacts causés par la ReLU, nous avons conçu un fantôme circulaire 2D homogène avec une région circulaire intérieure plus petite, comme le montre la figure 2.4. L’extérieur du fantôme a été fixé à 0 et le fond du fantôme à 0.25. Pour la région circulaire intérieure, des régions froides et chaudes ont été étudiées, avec des contrastes respectifs de 1:0,1 et 1:4 entre le fond du fantôme et la région étudiée. Une tâche de débruitage avec DIP (cf. partie 2.2.2) a été effectuée sur cette image avec l’architecture U-Net modifiée présentée dans la partie 2.3.3, avec ou sans la fonction d’activation ReLU à la fin de l’architecture. Plutôt que d’être vue comme un réel débruitage, cette tâche peut être considérée comme l’ajustement de la vérité terrain (notée GT par la suite) TEP par le DIP puisque celle-ci n’est pas bruitée. Pour les deux images GT considérées, une image de type anatomique a été utilisée comme entrée z du DIP, choisie comme l’image GT TEP avec insert froid ou celle avec insert chaud, donnant quatre cas différents à investiguer. Aucune mise à l’échelle n’a été utilisée pour les images d’entrée et d’étiquette. L’optimiseur utilisé est Adam, avec un taux d’apprentissage choisi manuellement à 0,01.

2.4.2 Simulations TEP 2D

2.4.2.1 Fantôme cérébral et simulations analytiques

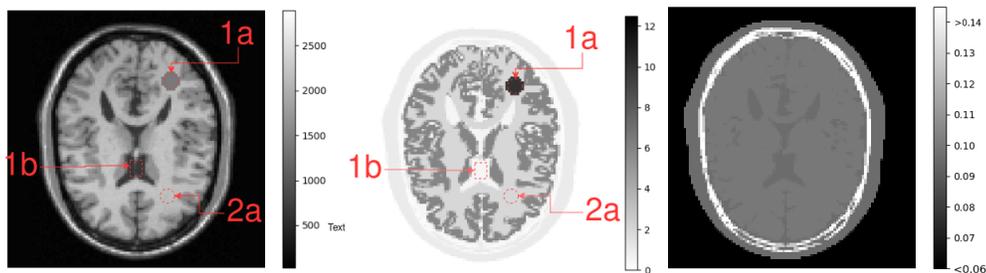


FIGURE 2.5 – Fantôme cérébral utilisé pour les simulations TEP FDG 2D (*gauche* : image IRM pondérée T1 ; *milieu* : GT TEP) ; *droite* : image d’atténuation en cm^{-1}).

Pour étudier les performances de l’algorithme, un fantôme cérébral 2D a été créé à l’aide de la base de données BrainWeb [23]. Une image IRM pondérée T1 a été simulée à l’aide de l’outil disponible sur le site Web. L’image de référence dans BrainWeb a été utilisée pour créer une GT TEP uniforme par morceaux, imitant une distribution cérébrale FDG classique avec les valeurs de fixation normalisée (Standardized Uptake Value - SUV) suivantes : matière grise à 6, matière blanche à 2, liquide céphalo-rachidien (LCR) à 0,5, os à 0,2 et autres parties à 1. Une tumeur circulaire uniforme a été insérée manuellement dans la matière blanche dans les images TEP

(SUV=10) et IRM. L'atténuation de la TEP a été fixée à $0,156\text{cm}^{-1}$ dans les os et à $0,096\text{cm}^{-1}$ ailleurs. Le fantôme a été échantillonné en 224×224 voxels de $1\text{ mm} \times 1\text{ mm}$. La figure 2.5 montre l'image IRM, la GT TEP et l'image d'atténuation.

Des simulations analytiques 2D TEP ont été réalisées à l'aide du logiciel décrit dans [87] et de la géométrie du scanner Biograph mMR de Siemens. Tout d'abord, les images d'atténuation et d'émission ont été sous-échantillonnées par un facteur de 2 et lissées par un filtre gaussien isotrope de 4 mm de largeur à mi-hauteur (Full Width at Half Maximum - FWHM) pour imiter la PSF du scanner [7]. Vingt réplicats statistiques de données non-TOF ont ensuite été simulées avec 40% de fraction de fortuites, 35% de fraction de diffusées et 5M de prompts.

2.4.2.2 Algorithmes de reconstruction

Plusieurs algorithmes ont été utilisés pour reconstruire les données simulées :

- l'algorithme DNA proposé,
- DIPRecon [37],
- ADMM-Reg [59] avec régularisation MRF quadratique (cf. partie 1.2.5.7.2),
- APPGML (qui s'est avéré donner le meilleur compromis biais-variance dans le cas de faibles statistiques d'acquisition dans l'étude de [62]) avec régularisation MRF quadratique (cf. partie 1.2.5.7.2),
- BSREM [1] avec la pénalité RDP [65] tel qu'il est utilisé dans la routine clinique (GE Healthcare),
- OSEM [52] tel qu'il est utilisé dans la routine clinique (Siemens Healthineers).

Les reconstructions ont été effectuées à l'aide du logiciel de reconstruction CASToR [61] (cf. Annexe C.1) . Pytorch lightning (<https://lightning.ai>) a été utilisé pour les étapes de débruitage avec le DIP. Les images ont été reconstruites en 112×112 voxels de $2\text{ mm} \times 2\text{ mm}$.

Pour le paramètre de pénalité ADMM ρ_2 spécifique au DNA, nous avons utilisé les politiques ρ_2 adaptatif et τ adaptatif [104], avec une valeur initiale ρ_2 de 1, $\mu_{residual} = 2$ et $\tau_{max} = 100$ (cf. partie 3.4.1).

Ensuite, à des fins de comparaison, les hyperparamètres communs à DNA et à DIPRecon ont été réglés avec les mêmes valeurs. Le paramètre de pénalité ADMM ρ_1 lié à la contrainte du DIP a été fixé manuellement à 3 (cf. partie 3.5.3.3). L'optimisation du DIP a été réalisée avec l'optimiseur Adam (cf. partie 3.5.1.1). Le taux d'apprentissage a été fixé expérimentalement à 0,01. Un paramètre de régularisation de décroissance des poids (weight decay en anglais) de 5×10^{-8} a été utilisé.

L'image BSREM de départ utilisée pour le DNA et le DIPRecon a été calculée en utilisant 30 itérations et 28 sous-ensembles, avec RDP et un paramètre de pénalité de $\beta = 0,5$. Elle est débruitée par le DIP, avec arrêt avant convergence grâce à l'algorithme EMV utilisant un facteur de décroissance $\alpha_{EMV} = 0,251$ et un nombre de patience $P = 200$ (cf. partie 3.3.2.2.2). Ensuite, pour chaque itération externe, 200 itérations du DIP ont été réalisées. Afin de stabiliser l'apprentissage du réseau pendant la reconstruction, l'entrée z du DIP a été normalisée en la divisant par son maximum [37] (normalisation au maximum, cf. partie 3.5.2). Quant à l'image courante à débruiter $x^{n+1} + \mu^n$, elle a aussi été mise à l'échelle à chaque itération externe n par une normalisation au maximum.

Seuls deux hyperparamètres ont été définis différemment entre DNA et DIPRecon en raison des différences intrinsèques entre les deux algorithmes. Premièrement, le nombre d'itérations pour le transfert d'optimisation (2.4a) du DIPRecon a été fixé à 2 comme dans [37], alors que 3 itérations internes ont été choisies expérimentalement pour l'optimisation du bloc bleu dans la Fig. 2.2. Deuxièmement, 750 et 500 itérations externes ont été effectuées pour le DNA et le DIPRecon respectivement.

Quant aux autres algorithmes, OSEM a été exécuté avec 36 itérations et 28 sous-ensembles et un filtre gaussien isotrope sur une gamme de FWHM de 0 à 15 mm a été appliqué après la reconstruction. BSREM a d'abord été utilisé avec un paramétrage proche de celui utilisé en routine clinique [2], avec 30 itérations et 28 sous-ensembles et pénalité RDP (le paramètre de pénalité a été choisi tel que $\beta = 2$ pour obtenir une image de bonne qualité visuelle). Les trois algorithmes suivants ont utilisés la méthode des poids de Bowsler pour inclure une information anatomique, à des fins de comparaison avec le DIPRecon et le DNA. BSREM a de nouveau été

utilisé avec 30 itérations et 28 sous-ensembles et pénalité RDP. Le nombre d'itérations utilisé pour ADMM-Reg a été choisi pour atteindre la convergence ($\varepsilon = 0,001$ [12]) avec un critère d'arrêt s'appuyant sur les résidus [104]. APPGML a été exécuté avec 30 itérations et 28 sous-ensembles comme BSREM, avec une borne $A=-10$ correspondant en valeur absolue à la valeur SUV maximale dans le fantôme. Ces deux derniers algorithmes utilisent une pénalité MRF avec un potentiel quadratique. Des reconstructions avec plusieurs valeurs de paramètre de pénalité β ont été effectuées.

Remarque :

- pour toutes les reconstructions BSREM avec RDP : par défaut dans le logiciel de reconstruction CASToR, les hyperparamètres (cf. partie 1.2.5.5.3 et partie 1.2.5.7.2.2) sont choisis tel que $\alpha_0 = 1$, $\gamma_B = 0,2$ et $\gamma = 2$,
- pour les pénalités de type MRF (cf. partie 1.2.5.7.2), le logiciel CASToR répartit la fonction de régularisation R sur les M sous-fonctions de coût Φ_m [1]. Dans cette étude, les pénalités MRF ont été utilisées avec un voisinage sphérique de 9 mm et des facteurs de proximité selon l'inverse de la distance euclidienne. De plus, les algorithmes APPGML, ADMM-Reg et BSREM + pénalité MRF avec poids de Bowsheer ont utilisé la version asymétrique (cf. partie 1.2.5.7.2.3) qui est implémentée dans le logiciel CASToR. Dans cette étude, les poids de Bowsheer ont été calculés à partir de l'image IRM avec un pourcentage $p = 40\%$ de voxels les plus similaires,
- l'initialisation DIP-BSREM utilisée pour initialiser DIPRecon diffère de celle utilisée pour le DNA uniquement par l'utilisation de la ReLU finale dans l'architecture du DIP.

2.4.2.3 Métriques d'évaluation

Pour évaluer les performances des algorithmes, nous avons défini des régions d'intérêt (Regions Of Interest en anglais - ROI), toutes uniformes dans cette étude.

Nous avons tout d'abord évalué la récupération d'activité dans la ROI de la tumeur. La récupération d'activité est calculé en pourcentage défini par la formule suivante :

$$\text{AR}(\text{ROI}) = \frac{m_{\text{ROI}}}{\text{GT}(\text{ROI})}, \quad (2.24)$$

où $m_{\text{ROI}} = \frac{1}{L} \sum_{l \in \text{ROI}} x_l$ la valeur moyenne de l'image x dans les L voxels de la ROI et $\text{GT}(\text{ROI})$ la valeur de la GT dans la ROI.

Ensuite, nous avons déterminé le biais relatif dans une région froide, ici la LCR, et dans une zone circulaire au sein de la matière blanche (région 2a de la figure 2.5). Le biais relatif est défini en pourcentage par la formule suivante :

$$\text{bias}_{\text{rel}}(\text{ROI}) = \frac{m_{\text{ROI}} - \text{GT}(\text{ROI})}{\text{GT}(\text{ROI})}, \quad (2.25)$$

En plus de ces mesures locales, le SSIM (Structural SIMilarity index) a été calculé pour effectuer une mesure globale dans l'ensemble du cerveau. Le MSSIM (Mean SSIM) mesure la similarité entre deux images en termes de luminance, de contraste et de structure [101]. On définit une fenêtre locale permettant de ne considérer qu'une partie des deux images d'intérêt pour calculer localement leur similarité. Pour calculer le MSSIM, cette fenêtre locale est déplacée sur les deux images. Nous avons choisi le paramétrage par défaut de la bibliothèque scikit-learn de Python. La formule suivante montre le SSIM entre les images fenêtrées A et B , avec les mêmes pondérations sur les termes de luminance, de contraste et de structure [101] :

$$\text{SSIM}(A, B) = \frac{(2m_A m_B + C_1)(2\sigma_{AB} + C_2)}{(m_A^2 + m_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}, \quad (2.26)$$

avec m_A , m_B , et σ_A , σ_B , respectivement la moyenne et l'écart-type des valeurs de voxels des images fenêtrées A et B , et σ_{AB} la covariance entre A et B . Par souci de simplicité, nous utiliserons dans la suite le terme SSIM au lieu de MSSIM, qui doit être compris comme la moyenne des similarités locales entre deux images.

Enfin, pour l'évaluation quantitative du niveau de bruit, nous avons utilisé la métrique de rugosité de l'image (IR - Image Roughness) dans le fond du fantôme. Elle mesure l'écart-type des valeurs des voxels du fond du fantôme par rapport à leur valeur moyenne. Elle est définie comme suit :

$$\text{IR}(\text{ROI}) = \frac{\sqrt{\frac{1}{L-1} \sum_{l \in \text{ROI}} (x_l - m_{\text{ROI}})^2}}{m_{\text{ROI}}}, \quad (2.27)$$

avec ROI étant le fond du fantôme, assigné ici à la matière blanche du fantôme, à laquelle a été appliquée une opération morphologique d'érosion. Cela permet de ne considérer que les voxels complètement intérieurs à la matière blanche.

Pour présenter les résultats, des courbes de compromis entre une des métriques précédemment énoncée et la rugosité de l'image ont été réalisées. Ces courbes seront tracées à partir des différentes valeurs :

- d'itérations externes pour les algorithmes DIPRecon et DNA,
- de paramètre de pénalité pour les algorithmes BSREM, APPGML et ADMM-Reg,
- de FWHM pour le post filtrage appliqué sur les images reconstruites OSEM.

2.4.3 Données réelles TEP 3D

Pour effectuer une reconstruction d'une image réelle à partir des données acquises sur la machine TEP/IRM du CHU de Nantes, il faut tout d'abord récupérer le sinogramme de prompts. Les sinogrammes d'atténuation, de normalisation, de coïncidences diffusées et fortuites sont calculés par le système, et serviront à effectuer toutes les corrections associées. Après l'utilisation d'un convertisseur spécifique à CASToR, ces sinogrammes sont disposés dans un fichier de données que le logiciel CASToR peut lire.

Les données brutes d'un examen cérébral réalisé sur le scanner TEP/IRM Biograph mMR installé au CHU de Nantes ont été utilisées. Le patient a reçu une injection de 171 MBq de FDG pour une acquisition d'une durée de 15 minutes, survenant 60 minutes après l'injection. Les images ont été reconstruites en $172 \times 172 \times 127$ voxels de $2,09 \text{ mm} \times 2,09 \text{ mm} \times 2,03 \text{ mm}$. Une PSF isotrope dans l'espace image avec un noyau gaussien de $4,5 \text{ mm}$ a été utilisée. L'image IRM pondérée T1 a été rééchantillonnée pour que la taille des voxels correspondent à ceux de l'image TEP.

L'algorithme DNA a été comparé à l'algorithme DIPRecon et à l'algorithme BSREM avec RDP. Pour BSREM, des paramètres proche de ceux utilisés en routine clinique (30 itérations, 28 sous-ensembles) [2] ont été choisis, avec un paramètre de pénalité $\beta = 0,001$ choisi expérimentalement. L'algorithme BSREM a également été exécuté avec RDP et poids de Bowsher à des fins de comparaison (30 itérations, 28 sous-ensembles, $\beta = 0,003$ choisi expérimentalement).

Pour les algorithmes DNA et DIPRecon, l'image BSREM de départ a été calculée en utilisant 30 itérations et 28 sous-ensembles avec RDP, et un paramètre de pénalité inférieure à celui de l'image clinique afin d'obtenir un meilleur contraste mais avec un bruit plus élevé ($\beta = 3 \times 10^{-6}$). Les valeurs des paramètres suivants ont été fixées différemment par rapport aux simulations 2D :

- Le nombre de patience de l'algorithme EMV a été fixé à $P = 1000$ et le nombre d'itérations du DIP à 600 pour les deux algorithmes,
- Pour le DNA, le paramètre de pénalité d'ADMM a été fixé à $\rho_1 = 3 \times 10^{-8}$ ainsi que 50 itérations internes. Pour le DIPRecon, nous avons choisi 2 itérations internes comme en 2D, et $\rho_1 = 3 \times 10^{-5}$.

Pour expliquer ces changements, chaque hyperparamètre lié au nombre d'itérations du DIP a été augmenté car il a été observé que le DIP était plus lent à débruiter une image dans ce contexte par rapport au cas 2D. En outre, le paramètre de pénalité d'ADMM ρ_1 et le nombre d'itérations internes ont été modifiés pour accélérer l'évolution de l'image reconstruite par le DNA. 100 itérations externes ont été effectuées pour obtenir une image avec un niveau de bruit qualitativement proche des images BSREM cliniques. Comme pour DIPRecon, nous avons choisi un paramètre de pénalité ADMM plus élevé car nous avons observé qu'autrement, le bruit augmentait trop rapidement après seulement quelques itérations externes. Quant à DIPRecon,

il a été exécuté pour 25 itérations externes. Le taux d'apprentissage et le facteur de décroissance α_{EMV} sont les mêmes que dans les simulations 2D. Enfin, pour les deux algorithmes, nous avons choisi de standardiser (moyenne 0 et variance 1, cf. partie 3.5.2) l'image en entrée du DIP et l'image bruitée à débruiter. En effet, nous avons observé expérimentalement que moins d'itérations étaient nécessaires pour débruiter une image avec cette méthode que la normalisation au maximum utilisée dans le cas 2D.

La prochaine partie présente les résultats obtenus avec le DNA et les autres algorithmes, pour les trois expériences décrites précédemment.

2.5 Résultats

2.5.1 Expérience DIP ReLU

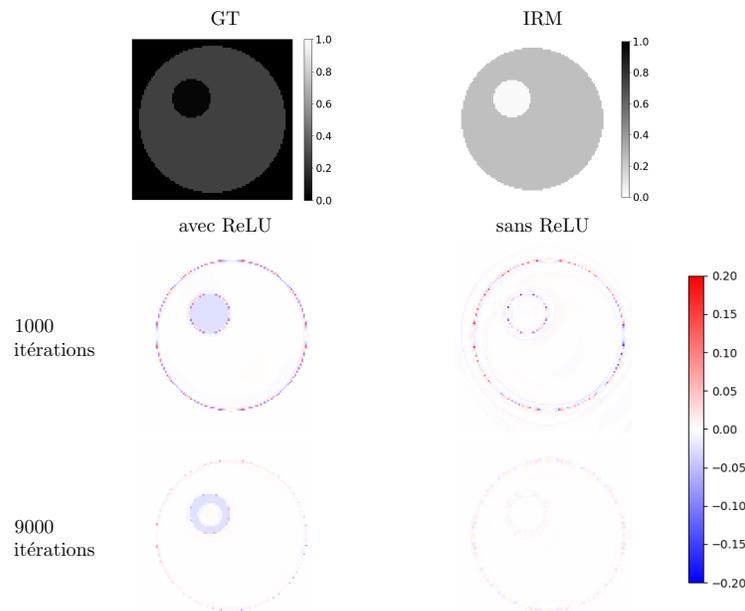


FIGURE 2.6 – Expérience DIP ReLU : reproduire une image TEP (première ligne à droite) avec le DIP prenant en entrée une image anatomique (première ligne à gauche). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (colonne de gauche) et sans ReLU (colonne de droite), après 1000 itérations (deuxième ligne) et 9000 itérations (troisième ligne), lors de l'ajustement de l'image TEP avec l'insert froid. L'image anatomique contient une zone en hyposignal.

La figure 2.6 montre les images de biais (sortie du DIP moins GT) obtenues pour l'expérience DIP ReLU, avec un réseau de neurones DIP avec ou sans la fonction d'activation ReLU finale, pour 1000 et 9000 itérations. Nous pouvons observer que l'insert froid n'est pas biaisé dans le cas de l'architecture sans ReLU pour 1000 et 9000 itérations, contrairement à la sortie du réseau de neurones avec ReLU après 1000 itérations, qui est clairement biaisée. Sans ReLU, la valeur moyenne dans l'insert froid est cohérente avec la GT, alors que la majorité des pixels ont été mis à 0 lorsque la ReLU était présente. Après 9000 itérations avec la ReLU, l'insert froid se rapproche de la valeur réelle, même si un grand nombre de pixels restent bloqués à 0.

Les trois autres configurations et résultats associés sont disponibles et commentés en annexe D.1. En résumé :

- lorsque le DIP cherche à reproduire la GT TEP contenant un insert chaud avec l'image anatomique ayant une zone en hypersignal, les deux architectures arrivent à reproduire la GT TEP après 9000 itérations,
- dans le cas où la GT TEP contient un insert chaud et l'image anatomique contient une zone en hyposignal, les deux architectures ont besoin de plus d'itérations pour reproduire la GT TEP, en particulier l'architecture avec ReLU,
- dans le cas où la GT TEP contient un insert froid et l'image anatomique contient une zone en hypersignal, les deux architectures ont besoin de plus d'itérations pour reproduire la GT TEP. Cependant, l'architecture sans ReLU réussit à reproduire la GT TEP après 9000 itérations, contrairement à l'architecture avec ReLU qui continue à produire des zéros au niveau de l'insert.

Nous avons également observés les mêmes artefacts sur l'architecture originale proposée pour le DIP dans [92], lorsque l'activation finale était une sigmoïde au lieu de la ReLU (résultats non montrés dans ce manuscrit).

2.5.2 Simulations 2D TEP

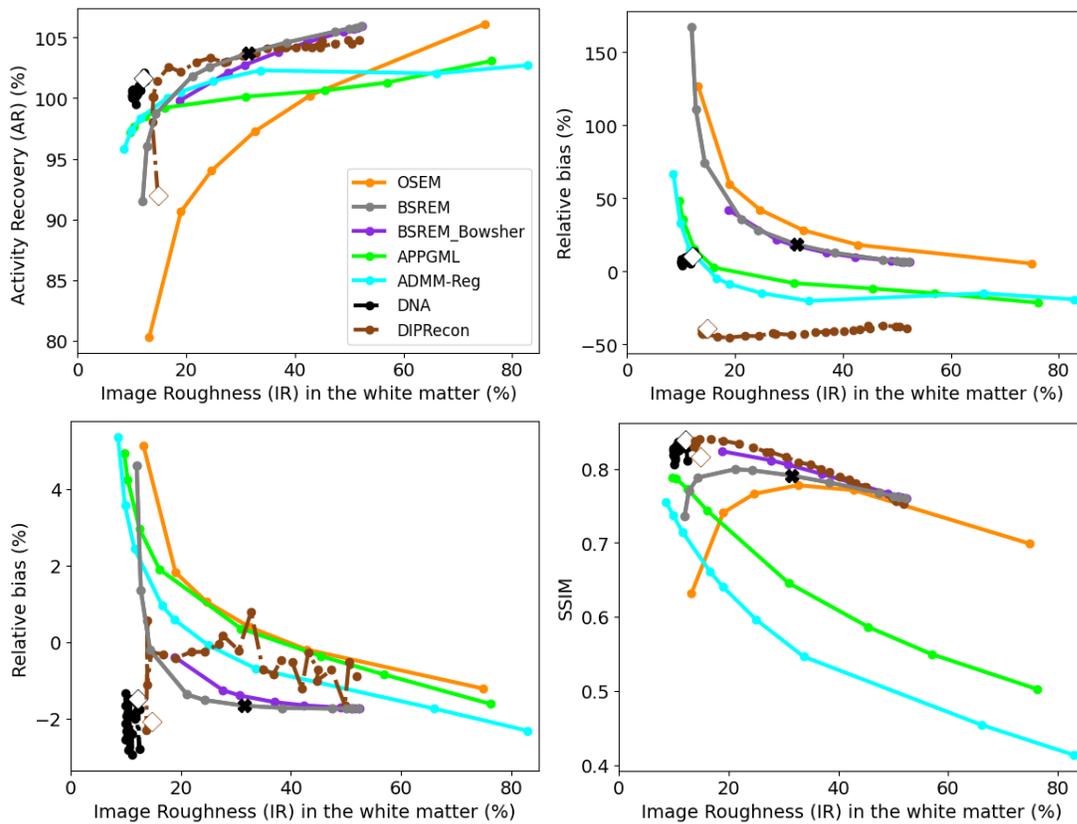


FIGURE 2.7 – Courbes de compromis moyennées sur 20 réplicats de la récupération d'activité dans la ROI tumeur (*haut gauche*), du biais relatif dans la ROI LCR (*haut droite*), du biais relatif dans la ROI matière blanche (*bas gauche*) et du SSIM dans l'ensemble du cerveau (*bas droite*), en fonction de la rugosité dans l'ensemble de la matière blanche, pour l'algorithme DNA proposé, DIPRecon, BSREM + RDP avec et sans pondérations de Bowsher, OSEM, APPGML et ADMM-Reg avec pénalité MRF quadratique et pondérations de Bowsher. Le BSREM de départ utilisé pour DIPRecon et DNA correspond à la croix noire. Les diamants blancs correspondent à cette image débruitée par le DIP (avec et sans ReLU) à l'aide de l'algorithme d'arrêt précoce EMV.

La figure 2.7 montre la récupération d'activité dans la zone d'intérêt de la tumeur, le biais

relatif dans les zones d'intérêt du LCR et de la matière blanche, et le SSIM dans l'ensemble du cerveau en fonction de la rugosité de l'image dans la matière blanche pour tous les algorithmes comparés.

Dans la zone d'intérêt tumorale (figure 2.7 en haut à gauche), toutes les méthodes n'utilisant pas le DIP ont une récupération d'activité proche de 100 % pour des valeurs de rugosité d'image élevées. DIPRecon montre une tendance similaire dans cette région. Pour l'algorithme DNA, le point d'initialisation est déjà très proche de 100 % pour une rugosité d'image plus faible que pour toutes les autres méthodes. La rugosité de l'image a d'abord diminué jusqu'à 300 itérations externes, puis a augmenté au fil des itérations, pour une récupération d'activité stable et très proche de 100 %.

Pour toutes les méthodes, le biais relatif dans la ROI de la matière blanche (figure 2.7 en bas à gauche) est proche de zéro (si le paramètre de pénalité n'est pas trop élevée pour les algorithmes n'utilisant pas le DIP).

En ce qui concerne le biais dans la zone d'intérêt du LCR (figure 2.7 en haut à droite), OSEM et BSREM surestiment l'activité, tandis qu'APPGML et ADMM-Reg se rapprochent de zéro lorsque la rugosité de l'image augmente. Le DIPRecon présente un fort biais négatif avec une valeur moyenne proche de zéro. Le DNA obtient une rugosité d'image plus faible que les autres algorithmes pour un biais très faible, s'améliorant au fil des itérations.

Les meilleures valeurs de SSIM ont été obtenues par DNA, DIPRecon et BSREM+RDP avec les poids de Bowsher (figure 2.7 en bas à droite). Pour le DNA, le diamant correspondant à l'initialisation du DNA a presque atteint le meilleur SSIM de la courbe du DNA. La reconstruction avec plusieurs itérations du DNA ne semble pas améliorer le SSIM. Les mêmes observations ont été faites pour l'algorithme DIPRecon.

Nous observons également que pour le même ensemble d'hyperparamètres entre DIPRecon et DNA, la rugosité de l'image évolue plus rapidement au fil des itérations pour le DIPRecon que pour le DNA (Fig. 2.7).

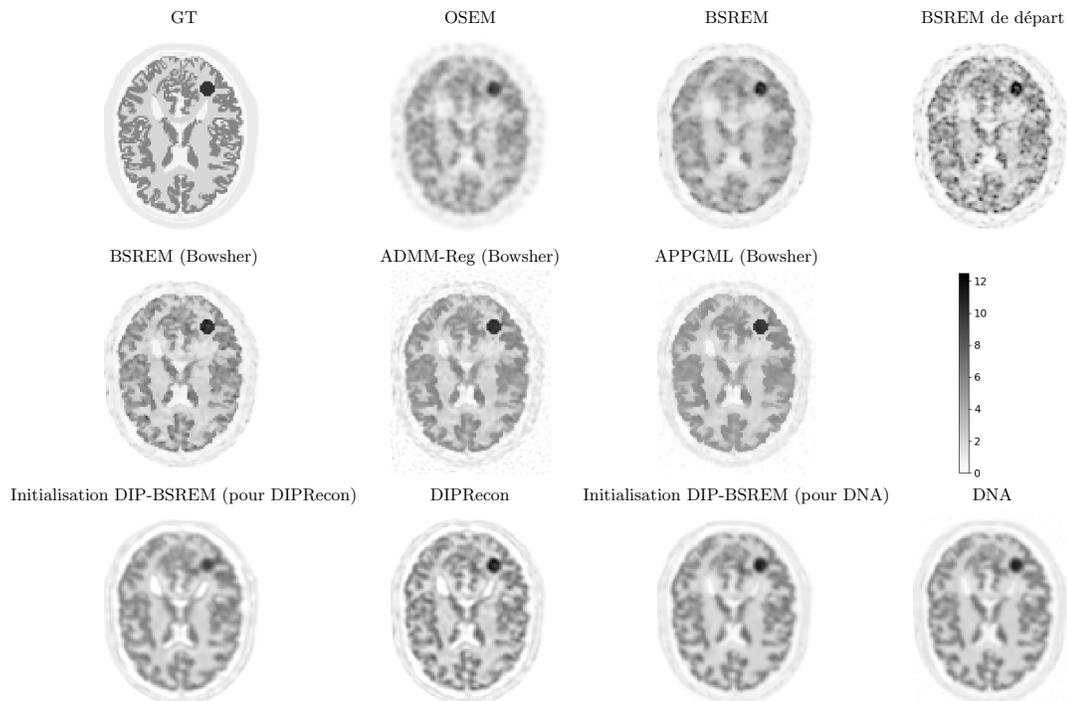


FIGURE 2.8 – Images entre 10 et 25 % de rugosité pour tous les algorithmes calculés. L'image BSREM de départ et les initialisations DIP-BSREM utilisées pour initialiser le DIPRecon et le DNA sont aussi présentées.

La figure 2.8 montre les images reconstruites pour un répliat et pour une rugosité d'image

similaire pour tous les algorithmes. Comme la GT TEP et l'image IRM partagent des structures communes, les méthodes utilisant l'information IRM fournissent plus de détails que les autres (OSEM et BSREM). De plus, les méthodes s'appuyant sur le DIP semblent intégrer l'information IRM plus naturellement que celles utilisant les poids de Bowsher. Les initialisations DIP-BSREM utilisées pour le DIPRecon et DNA sont assez proches, bien que celle du DIPRecon montre des régions froides plus claires et entourées de bords plus sombres, non visibles dans l'initialisation du DNA ou la GT. Les images confirment aussi la moins bonne quantification au niveau de la tumeur pour l'initialisation du DIPRecon par rapport à celle du DNA. De plus, l'image reconstruite par le DNA est très proche de l'initialisation DIP-BSREM, alors qu'on voit des détails plus marqués pour le DIPRecon par rapport à son initialisation. Tout comme l'initialisation, l'image DIPRecon comporte des régions froides entourées de bords sombres qui ne sont pas présents dans l'image DNA et la GT. Le biais négatif du DIPRecon est aussi visible dans le LCR qui apparaît plus blanc qu'avec les autres méthodes. Enfin, l'image produite par le DNA est plus lisse et n'est pas biaisée dans les régions froides.

2.5.3 Données réelles 3D TEP

La figure 2.9 montre une coupe des images reconstruites du scanner cérébral FDG 3D, avec BSREM + RDP avec ou sans poids de Bowsher, le BSREM de départ et son débruitage par le DIP, ainsi que les images reconstruites avec DNA et DIPRecon. Tout comme les images 2D, l'utilisation de l'information IRM avec le DIP ajoute beaucoup de structures pour le DIPRecon et le DNA, et semble le faire de manière plus naturelle qu'avec la méthode de Bowsher. Les initialisations DIP-BSREM du DIPRecon et du DNA ont capturé les détails de l'image IRM aussi clairement que les images reconstruites avec ces deux algorithmes. Cependant, le DNA a mieux réussi à reconstruire les régions chaudes apparaissant uniquement en TEP, ce qui la rend plus proche de l'image BSREM clinique. En outre, le DIPRecon montre des structures légèrement meilleures que le DNA, bien qu'il ait également exacerbé certaines régions avec des concentrations d'activité très intenses. Nous observons également que la cavité nasale apparaît plus blanche avec le DNA et le BSREM Bowsher qu'avec le BSREM de départ et le DIPRecon.

2.6 Discussion

Dans ce travail, l'algorithme DNA a été développé comme une amélioration de l'algorithme DIPRecon afin de supprimer ses biais intrinsèques dus à l'utilisation de la fonction d'activation ReLU finale dans le réseau de neurones. Nous avons observé une amélioration de la région chaude avec l'algorithme DNA par rapport à l'algorithme DIPRecon (Fig. 2.7 en haut à gauche). En outre, la région froide a été bien récupérée par l'algorithme DNA, tandis que le DIPRecon a été biaisé négativement (figure 2.7 en haut à droite) parce qu'il a souffert de la ReLU qui a mis la majorité des pixels à zéro (étant donné que la valeur GT était légèrement positive dans cette région).

D'après les résultats de l'expérience DIP ReLU (section 2.5.1), les pixels de sortie du DIP pouvaient être mis à zéro et bloqués à zéro en raison de la fonction d'activation ReLU finale. Pendant l'optimisation du DIP, certains neurones de la dernière couche peuvent devenir négatifs. Le gradient devient alors nul ce qui ne permet plus au poids d'être mis à jour avant la ReLU finale, affectant la bonne valeur de certains pixels de l'image de sortie. Les artefacts dus à la ReLU peuvent même devenir si importants que l'image entière est remplie de zéros. C'est ce qui s'est produit dans les simulations 2D pour un réplicat utilisant l'algorithme DIPRecon. Ce réplicat a finalement été supprimé des résultats de la partie 2.5.2.

Pour les reconstructions 3D avec DIPRecon et DNA, nous avons décidé d'utiliser la mise à l'échelle de standardisation plutôt que la normalisation au maximum comme pour les reconstructions 2D. À première vue, cela pourrait sembler préjudiciable à DIPRecon en raison de la contrainte de non-négativité en sortie du DIP. Cependant, lors de l'utilisation de la normalisation au maximum, les artefacts dus à la ReLU ont pris tellement d'importance que l'image reconstruite par DIPRecon est devenue complètement nulle après quelques itérations externes. En revanche, avec la standardisation, nous avons observé que les artefacts dus à la ReLU n'étaient pas

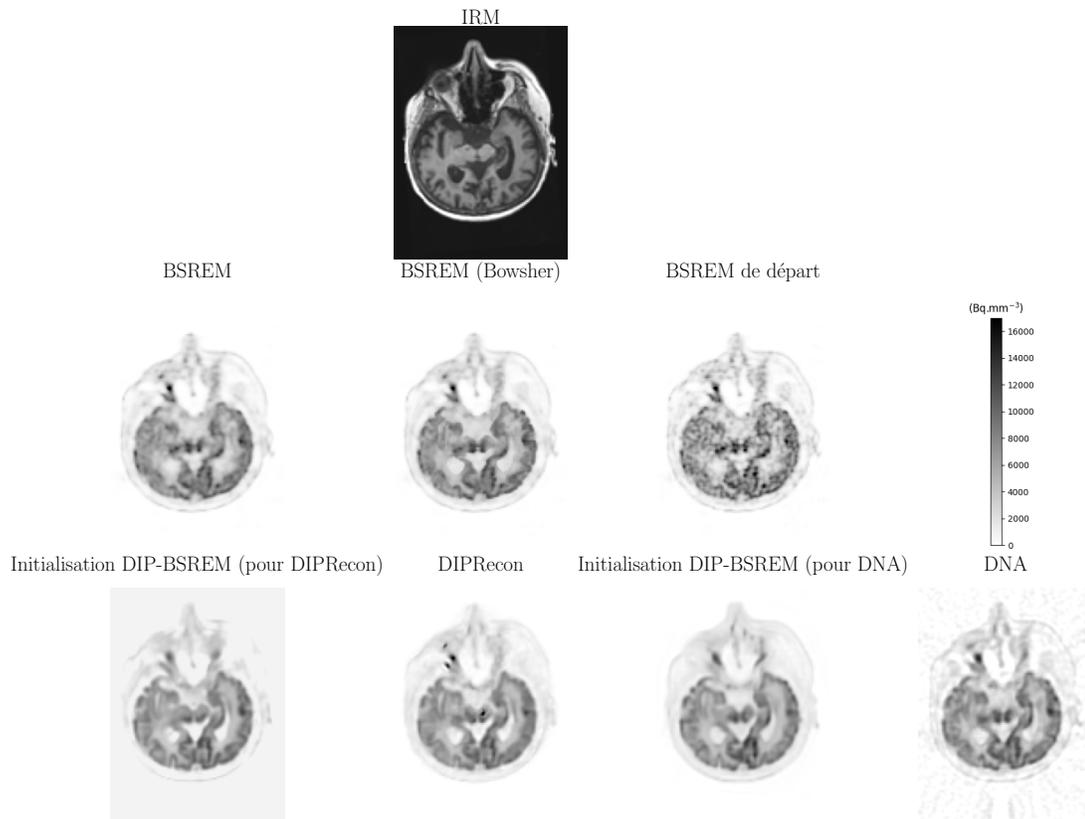


FIGURE 2.9 – Coupe axiale des images reconstruites à partir des données réelles 3D. L'image IRM T1 est représentée sur la *première ligne*. L'image BSREM clinique (avec et sans les poids de Bowsher) ainsi que l'image de départ BSREM sont représentées sur la *deuxième ligne*. Les initialisations DIP-BSREM utilisées pour initialiser le DIPRecon et le DNA, ainsi que les images reconstruites par DIPRecon et DNA sont indiquées sur la *dernière ligne*.

présents dans l'image reconstruite par DIPRecon. Mais un biais positif a été ajouté à l'extérieur du fantôme et dans les régions froides (voir Fig. 2.9).

Le problème d'optimisation que cherche à résoudre l'algorithme DNA n'est pas convexe à cause de la présence du réseau de neurones (cf. partie 2.3.3). De plus, l'algorithme ADMM est initialement proposé pour résoudre un problème d'optimisation sous contrainte linéaire, ce qui n'est le cas ici à cause de la non linéarité du réseau de neurones. Le DNA ne possède donc pas de propriétés théoriques de convergence. Cependant, comme les poids du DIP à une itération n sont utilisés pour initialiser l'entraînement du DIP à l'itération $n+1$, le DIP finit par entrer dans son régime de surapprentissage. Ainsi, après un grand nombre d'itérations, l'algorithme DNA produit des images de l'algorithme ADMM-Reg sans régularisation. Par conséquent, au cours des itérations, l'algorithme DNA peut être considéré comme un chemin entre son initialisation, ici l'image BSREM de départ débruitée par le DIP, et une image ADMM-Reg convergée sans aucune régularisation. Cette observation est également valable pour l'algorithme DIPRecon, qui aboutirait finalement à l'algorithme MLEM.

Dans cette étude, l'algorithme DNA a été initialisé avec une image BSREM de départ débruitée par le DIP à l'aide de l'algorithme EMV. Cette image présentait déjà d'assez bonnes propriétés à la fois qualitativement (Fig. 2.8) et quantitativement (Fig. 2.7). Après cette initialisation, l'effet principal du DNA est double. Le premier effet est une amélioration des métriques, en particulier dans les régions froides où l'image BSREM de départ est intrinsèquement biaisée. L'autre effet est visible dans la reconstruction 3D : certains détails de l'image BSREM de départ spécifiques aux données TEP étaient plus ou moins perdus lors du débruitage DIP initiale à l'aide de l'image anatomique. Mais ces détails ont été récupérés après plusieurs itérations du DNA. Dans [37], les auteurs ont montré que l'utilisation de DIPRecon était également utile pour que l'image reconstruite ait de meilleures structures que l'image d'initialisation. Nous supposons que ce n'était pas le cas dans nos expériences car [37] utilise un nombre fixe d'itérations pour débruiter leur image de départ, ce qui peut donner une image différente de celle obtenue avec l'algorithme EMV. En outre, cela pourrait expliquer la diminution du SSIM dans la figure 2.7 pour DIPRecon et DNA. En effet, pour passer d'une image d'initialisation relativement optimisée à l'image ADMM-Reg non-régularisé (pour DNA) ou l'image MLEM (pour DIPRecon), il est nécessaire de supprimer certaines structures apportées par l'image IRM.

Afin d'évaluer quantitativement les performances de DIPRecon et du DNA dans les simulations, leurs hyperparamètres communs ont été fixés aux mêmes valeurs. Il a été constaté que le DNA aurait besoin de plus d'itérations pour converger vers l'optimum ADMM-Reg comparé au DIPRecon pour atteindre l'optimum MLEM (voir figure 2.7). Cela s'explique par le fait qu'ADMM-Reg converge plus lentement que le transfert d'optimisation du DIPRecon. Cette observation tirée des simulations 2D explique la décision d'utiliser un paramètre de pénalité ADMM ρ_1 plus faible pour DNA que pour DIPRecon lors des reconstructions 3D. En outre, le taux d'apprentissage et le nombre d'itérations externes ont été sélectionnés manuellement dans cette étude. Tous ces hyperparamètres pourraient avoir un impact significatif sur les images produites par DNA et DIPRecon. Pour une analyse plus approfondie, des méthodes doivent être trouvées pour les sélectionner automatiquement. Certains hyperparamètres ont été regardés en détail dans le chapitre 3.

Pour les travaux futurs, la comparaison entre le DNA et les algorithmes de reconstruction de bout en bout avec DIP, comme proposé par [47] [46] [81] [53] pourrait être pertinente. D'une part, cette classe de méthodes a donné des résultats comparables à DIPRecon [47], voire de meilleurs résultats en ajoutant une pénalité explicite au DIP [46] [81] [53]. Ils sont plus rapides pour reconstruire une image et nécessitent moins d'hyperparamètres à régler. D'autre part, l'algorithme EMV pourrait potentiellement être appliqué à cette classe d'algorithmes. L'ajout de la contrainte de non-négativité pourrait également être effectué en dérivant un unique algorithme ADMM au lieu d'un ADMM imbriqué comme pour DNA.

En outre, de nombreuses idées ou méthodes peuvent être empruntées à d'autres articles et utilisées ou adaptées à l'algorithme proposé. Pour stabiliser l'apprentissage du DIP, le gradient peut être seuillé [69] pour éviter son explosion. L'utilisation d'une moyenne glissante exponentielle sur les sorties du réseau DIP comme dans [69] [88] pourrait aussi permettre d'obtenir des résultats de DIP plus stables. L'ajout d'un échantillon de bruit différent à l'entrée du DIP à

chaque itération pourrait également aider l'apprentissage du DIP à être plus robuste [107] [68] [18]. Le préapprentissage du DIP a également été proposé [24; 67]. Pour cela, les auteurs dans [24] ont construit une base de données TEP composée de paires d'images à faibles statistiques et à hautes statistiques, alors que l'apprentissage auto-supervisé (self-supervised learning en anglais) est utilisé dans [67]. Ces deux méthodes de préapprentissage améliorent les performances d'un débruitage avec un réseau DIP initialisé aléatoirement. Enfin, la méthode proposée dans ce chapitre pourrait également être étendue avec une régularisation explicite ou une estimation bayésienne pour pallier le problème de surapprentissage du DIP (cf. partie 3.6.4.2.2 pour des détails et commentaires sur ces méthodes).

L'algorithme DNA proposé dans ce chapitre peut avoir un intérêt pour l'imagerie TEP après radioembolisation à ^{90}Y . La partie suivante introduit cette application et présente des expériences à partir de simulations 2D.

2.7 Reconstruction sur des données ^{90}Y

2.7.1 Introduction du DNA dans le contexte de la radioembolisation à ^{90}Y

Cette partie est une étude complémentaire du DNA à une application de la radioembolisation avec microsphères d' ^{90}Y proposée au CHU de Nantes (cf. partie 1.3.3.2). La reconstruction dans ce contexte spécifique est un défi, car le facteur d'embranchement de ^{90}Y lié à l'émission d'un positon est très faible (cf. 1.3). En effet, le nombre de coïncidences vraies est tellement faible que celles-ci se retrouvent noyées dans les coïncidences fortuites naturellement générées par le signal de fond du LSO (cf. partie 1.1.4.2), générant une fraction de coïncidences fortuites autour de 90% en routine clinique. L'algorithme ADMM-Reg [59] a initialement été proposé dans ce contexte pour réduire le biais positif des algorithmes de reconstruction avec contrainte de non-négativité dans l'image (cf. partie 1.2.5.5.4). De même, les auteurs dans [62] ont proposé une étude comparative d'algorithmes de réduction de biais dans ce contexte spécifique.

Le DNA s'appuie mathématiquement sur l'algorithme ADMM-Reg, ayant montré des propriétés de réduction de biais dans ce contexte [59]. De plus, le DNA a démontré des propriétés intéressantes de réduction de biais dans le cas FDG, où le biais dans les régions froides est moins important que dans le cas ^{90}Y . Nous avons ainsi évalué l'algorithme DNA dans cette situation.

2.7.2 Simulations 2D ^{90}Y et paramètres des algorithmes de reconstruction

2.7.2.1 Fantôme ^{90}Y

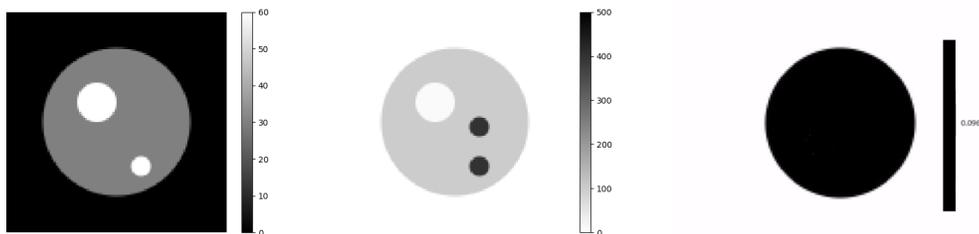


FIGURE 2.10 – Fantôme utilisé pour les simulations TEP 2D ^{90}Y (*gauche* : image anatomique ; *milieu* : GT TEP) ; *droite* : image d'atténuation en cm^{-1} .

Un fantôme cylindrique rempli d' ^{90}Y présenté en figure 2.10 comprenant une région froide

avec un contraste de 0,1:1 et deux régions chaudes avec un contraste de 4:1 a été utilisé. L'image d'atténuation a été choisie uniforme avec une valeur de $0,096\text{cm}^{-1}$. L'image anatomique a été définie à partir de l'image TEP, avec les mêmes bords que la région froide et l'une des régions chaudes, appelée région "TEP match". La deuxième région TEP est absente de l'image anatomique et sera appelée région "TEP seule". 112×112 voxels de $4\text{ mm} \times 4\text{ mm}$ ont été utilisés pour échantillonner le fantôme et pour les reconstructions. La PSF a été choisie comme dans la partie 2.4.2.2.

Tout comme les reconstructions du cerveau présentées précédemment, nous avons utilisé des simulations analytiques 2D [87] avec la géométrie du Siemens Biograph mMR TEP/IRM. 1,5M de coïncidences prompts ont été simulées avec une fraction de coïncidences fortuites de 90% et une fraction de coïncidences diffusées de 30%.

Remarque : Nous avons construit ce fantôme pour comprendre l'impact sur la reconstruction avec le DNA dans le cas d'une image anatomique utilisée en entrée du DIP, contenant une région chaude concordante avec la GT TEP, et une autre seulement présente dans la GT TEP. Ce cas de figure peut arriver en routine clinique dans une imagerie TEP après radioembolisation à ^{90}Y .

2.7.2.2 Algorithmes de reconstruction

Les algorithmes de reconstruction utilisés pour la comparaison sont les mêmes que dans la partie 2.4.2.2. Les paramètres de reconstruction associés sont les mêmes, exceptés des valeurs de β plus faibles pour tous les algorithmes, et une borne $A = -1000$ pour APPGML, adaptée à ce nouveau fantôme. De plus, les algorithmes APPGML et ADMM-Reg ont été utilisés avec une pénalité MRF quadratique sans facteur de similarité (i.e. sans les poids de Bowsher). Seul l'algorithme BSREM a été utilisé avec RDP avec ou sans poids de Bowsher.

Quant aux algorithmes DNA et DIPRecon, nous avons choisi la valeur de ρ_1 à 0,003 pour obtenir un comportement du DNA et du DIPRecon évoluant plus rapidement que dans la partie 2.5.2. De même, nous avons choisi 10 itérations internes pour le DNA pour accélérer sa convergence. Le nombre d'itérations du DIP a été choisi à 100 manuellement pour les deux méthodes. 300 itérations externes ont été réalisées pour les deux algorithmes. Les autres hyperparamètres ont été choisis comme dans la partie 2.4.2.2.

Les méthodes DNA et DIPRecon utilisent l'image anatomique en entrée du réseau DIP. L'algorithme BSREM+RDP avec poids de Bowsher utilise aussi l'image anatomique en tant qu'image de similarité pour déterminer les poids de la méthode de Bowsher.

Pour évaluer les performances quantitatives des algorithmes, nous avons tout d'abord évalué la récupération d'activité dans les deux régions chaudes ("TEP seule" et "TEP match"). Le bias relatif dans la zone froide ainsi que le SSIM dans l'ensemble du fantôme ont été calculés. Ces métriques ont été étudiées en fonction de la rugosité de l'image dans le fond du fantôme (ensemble des voxels dans le fantôme auxquels ont été retirés les voxels des trois régions).

Dans le contexte de la radioembolisation à ^{90}Y , le DNA doit être comparé en particulier avec ADMM-Reg et APPGML, étant conçus spécialement pour réaliser une réduction de biais dans ce contexte.

2.7.3 Résultats sur le fantôme ^{90}Y

La figure 2.11 présente les courbes de compromis obtenues dans les différentes régions d'intérêt et le SSIM global en fonction de la rugosité dans le fond du fantôme. Dans la région chaude "TEP match" (en haut à droite de la figure 2.11), l'initialisation du DIPRecon présente une moins bonne récupération d'activité que celle du DNA dans cette région. La courbe représentant la récupération d'activité de l'algorithme DIPRecon diminue légèrement pendant les premières itérations externes, puis augmente en se superposant quasiment à la courbe obtenue avec BSREM. L'algorithme DNA est quant à lui initialisé avec une image présentant une très bonne activité dans cette région, tout comme l'image de départ BSREM. L'initialisation du DNA a comme avantage de présenter un bruit beaucoup moins élevé par rapport à cette image BSREM, et de continuer à améliorer la récupération d'activité en autorisant une légère augmentation du bruit. Seul BSREM+RDP avec poids de Bowsher présente une récupération d'activité plus élevée

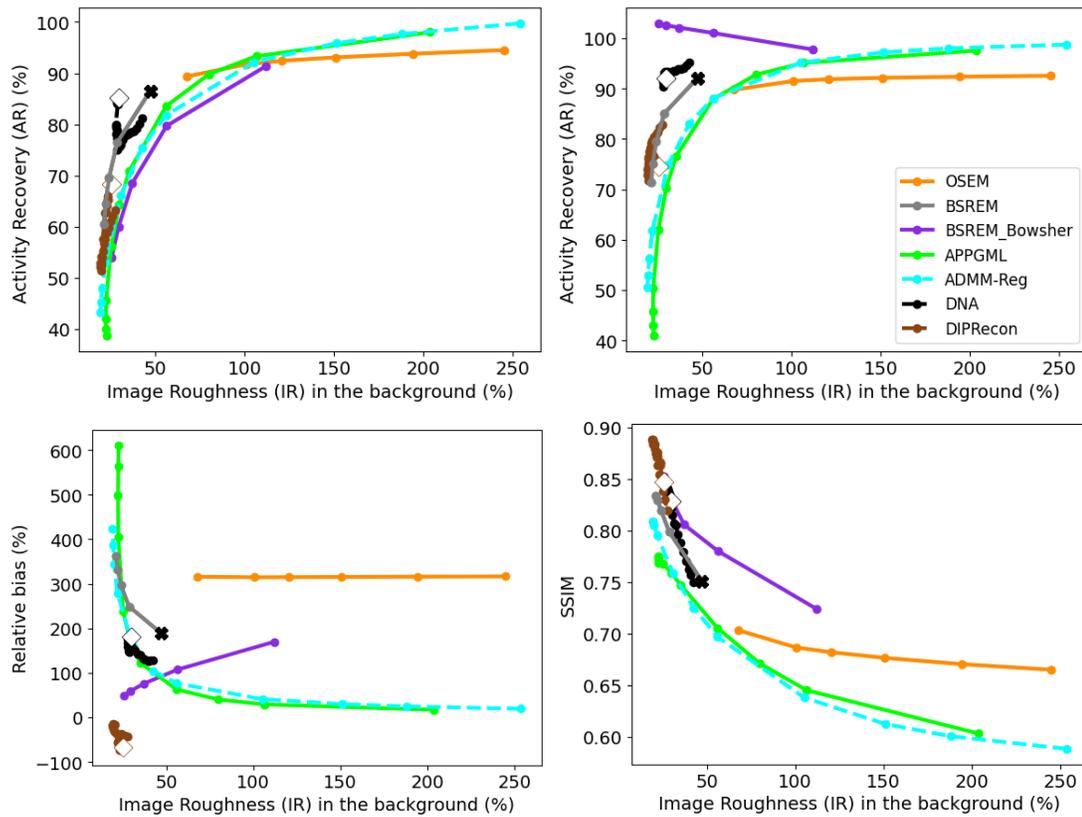


FIGURE 2.11 – Courbes de compromis moyennées sur 20 réplicats pour la récupération d’activité dans la ROI tumeur “TEP seule” (*haut gauche*) et la ROI tumeur “TEP match” (*haut droite*), le bias relatif dans la ROI de la zone froide (*bas gauche*) et le SSIM dans l’ensemble du fantôme (*bas droite*), en fonction de la rugosité dans le fond du fantôme, pour l’algorithme DNA proposé, DIPRecon, BSREM+RDP avec et sans poids de Bowsher, OSEM, et APPGML et ADMM-Reg avec pénalité MRF quadratique. Le BSREM de départ utilisé pour DIPRecon et DNA correspond à la croix noire. Les diamants blancs correspondent à cette image BSREM de départ débruitée par le DIP à l’aide de l’algorithme d’arrêt précoce EMV.

par rapport au DNA, et présente le meilleur compromis par rapport à APPGML, ADMM-Reg, OSEM dans cette région.

A propos de la zone chaude “TEP seule” (en haut à gauche de la figure 2.11), l’initialisation donnée à DIPRecon est encore une fois moins bonne que celle pour le DNA. La récupération d’activité des deux algorithmes diminue pendant les premières itérations, puis augmente avec une augmentation de la rugosité de l’image. L’algorithme DNA étant initialisé avec une image ayant une bonne récupération d’activité, il obtient un meilleur compromis que les autres algorithmes de réduction de biais. D’un autre côté, BSREM obtient une meilleure récupération d’activité que le DNA et le DIPRecon pour une même valeur de rugosité de l’image. L’algorithme BSREM+RDP avec poids de Bowsher présente cette fois-ci le moins bon compromis par rapport à APPGML, ADMM-Reg, OSEM, BSREM et le moins bon parmi les méthodes utilisant une image anatomique.

Dans la région froide (en bas à gauche de la figure 2.11), BSREM et OSEM présentent un biais positif plus élevé que dans la partie 2.5. ADMM-Reg et APPGML sont capables de diminuer le biais par rapport à BSREM et OSEM, et ne présentent quasiment plus de biais pour une rugosité de l’image très élevée. Quant à DIPRecon, son initialisation souffre d’un biais très négatif. Au cours des itérations, le biais négatif augmente, pour un bruit raisonnable. De plus, le DNA démarre avec le même biais que l’image BSREM de départ pour moins de bruit. Son biais diminue selon une courbe proche de ADMM-Reg et APPGML lorsque la rugosité de l’image augmente.

Enfin, ADMM-Reg, APPGML et OSEM présentent un SSIM inférieur à BSREM+RDP avec et sans poids de Bowsher et aux algorithmes utilisant le DIP (cf. en bas à droite de la figure 2.11). Le SSIM du DNA est maximal à son initialisation, puis diminue au cours des itérations externes. Quant au DIPRecon, les premières itérations externes permettent d’améliorer le SSIM en diminuant légèrement la rugosité de l’image. Le SSIM diminue ensuite avec une augmentation de la rugosité de l’image. Ces algorithmes présentent un meilleur SSIM que BSREM pour un niveau de bruit plutôt faible. BSREM+RDP avec poids de Bowsher présente un SSIM comparable aux initialisations de DNA et DIPRecon. Il est l’algorithme parmi ces derniers dont le SSIM diminue le moins vite quand la rugosité de l’image augmente.

La figure 2.12 montre les images reconstruites pour un répliat du fantôme ^{90}Y avec les différents algorithmes étudiés. Pour chaque algorithme, une image de la figure 2.12 a été choisie parmi toutes les images reconstruites après inspection visuelle.

L’image OSEM calculée après 2 itérations et 28 sous-ensembles semble peu convergée, avec des régions peu définies et peu contrastées. Comme vu dans les courbes de compromis, l’image BSREM de départ est plus bruitée mais présente un meilleur contraste dans les régions chaudes que l’image BSREM clinique. Les algorithmes de réduction de biais reconstruisent la zone froide avec moins de biais que les algorithmes cliniques ce qui est en accord avec les courbes de compromis. De plus, l’image utilisée à l’initialisation DIP-BSREM est fortement débruitée par rapport à l’image BSREM de départ. L’image reconstruite avec DNA après 200 itérations externes est plus bruitée que son initialisation, mais avec une région froide plus claire. Quant à l’image reconstruite avec DIPRecon, elle présente une très faible activité dans la région “TEP seule”, ainsi qu’une région froide qui est de plus grande taille que la GT, et des pixels valant zéro. Les trois images incluant du DIP montrent une région froide plus circulaire que les algorithmes n’utilisant pas d’information anatomique, mais présentent des contours chauds autour de la région froide. Enfin, l’image BSREM+RDP avec poids de Bowsher reconstruit extrêmement bien la région chaude “TEP match”, et assez fidèlement la région froide.

2.7.4 Discussion sur le fantôme ^{90}Y

D’une part, le DNA est plus performant que les algorithmes APPGML et ADMM-Reg dans les régions chaudes, en particulier dans la région “TEP match” avec information anatomique. L’apport d’information anatomique aide à récupérer des contours plus clairs de la zone “TEP match” par rapport à la zone “TEP seule”. La récupération d’activité du DNA est aussi meilleure que celle obtenue par APPGML et ADMM-Reg pour une même rugosité de l’image. L’initialisation DIP-BSREM permet d’initialiser le DNA avec une très bonne récupération d’activité, en

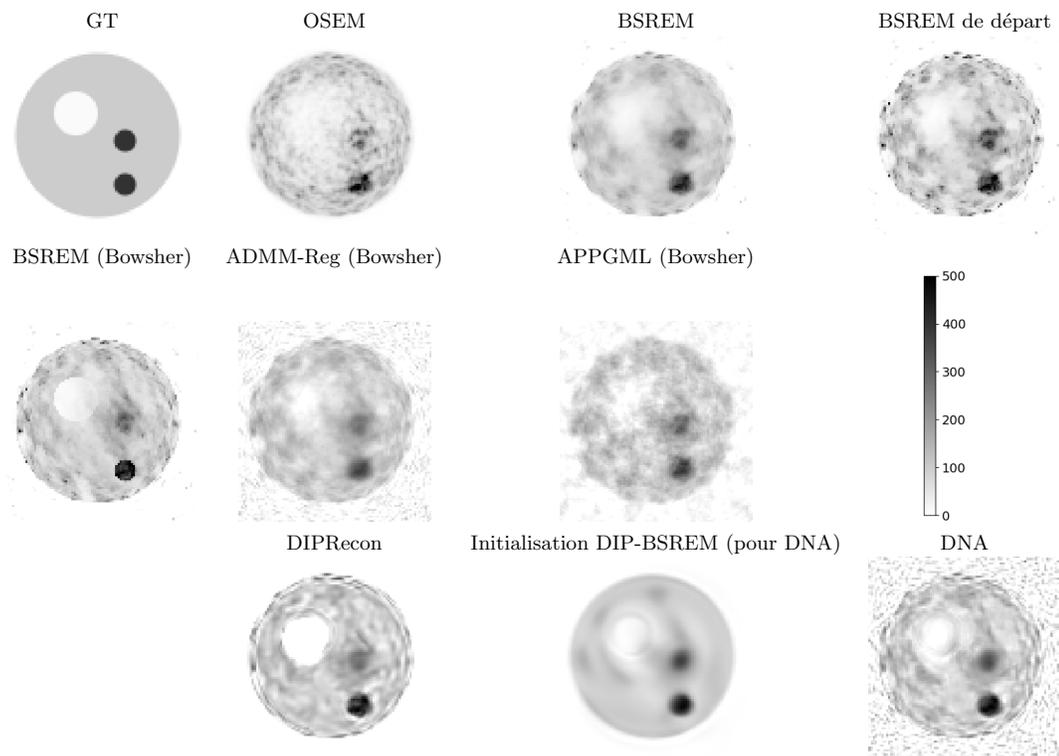


FIGURE 2.12 – Images reconstruites du fantôme ^{90}Y pour tous les algorithmes étudiés. L'image BSREM de départ et l'initialisation DIP-BSREM utilisée pour initialiser le DNA est aussi présentée. L'image OSEM présentée ici est montrée après 2 itérations et 28 sous-ensembles, avec un post filtrage gaussien avec FWHM de 4mm.

particulier dans la région avec information anatomique, tout en diminuant le bruit par rapport à l'image BSREM de départ.

L'apport d'information anatomique a aidé le DNA à conserver une bonne récupération d'activité tout au long des itérations. Cependant, lorsque celle-ci n'informe pas sur la position d'une zone de contraste en TEP, les performances dès l'initialisation du DNA sont moins bonnes et peuvent chuter dans les premières itérations. Ce phénomène est aussi visible pour le DIPRecon.

Quant aux algorithmes avec contrainte de non-négativité dans l'image, nous remarquons rapidement qu'ils sont biaisés positivement (sauf DIPRecon qui reconstruit beaucoup de voxels à zéro dans cette étude, ce qui n'est pas désirable). L'initialisation du DNA est aussi biaisée positivement dans la région froide, mais le DNA apporte une réduction de biais comparable à celle d'ADMM-Reg et APPGML pour un même niveau de bruit. En outre, l'algorithme DNA devrait converger vers la solution ADMM-Reg sans régularisation en raison du comportement de surapprentissage du DIP. Dans cette expérience, la courbe de compromis du DNA se rapprocherait de celle de l'algorithme ADMM-Reg dans un régime où la rugosité de l'image serait très élevée, et donc où l'image visuelle ne serait pas utilisable en pratique. Le DNA utilisant une image BSREM de départ peut ainsi être vu comme une amélioration de l'algorithme ADMM-Reg notamment dans les premières itérations.

Enfin, la méthode avec poids de Bowsher est très performante dans la région chaude avec information anatomique. Celle-ci guide de façon précise la forme et l'activité de la région à reconstruire. En revanche, sans information anatomique, la méthode devient la moins bonne de toute dans la région chaude. De plus, son comportement dans la zone froide est différent des autres algorithmes (le biais augmente en même temps que la rugosité, cf. figure 2.11 bas gauche), et nécessiterait une méthode pour choisir de façon optimale le paramètre de pénalité pour une bonne réduction de biais.

2.8 Conclusion

Dans ce travail, nous avons proposé le DNA, un nouvel algorithme de reconstruction TEP intégrant le DIP pour la réduction du bruit et une image anatomique. L'algorithme DNA a été développé comme une amélioration de l'algorithme DIPRecon proposé par [37] dans le but de supprimer ses biais intrinsèques résultant de la contrainte de non-négativité dans l'espace image. Pour cela, la ReLU finale a été supprimée et les valeurs négatives des voxels ont été autorisées en transférant la contrainte de non-négativité de l'espace image à l'espace de projection, comme proposé par [59]. L'algorithme DNA a abouti à des niveaux de biais inférieurs à DIPRecon et à d'autres algorithmes avec une contrainte positive intrinsèque sur l'image (OSEM et BSREM avec RDP), et similaires aux algorithmes autorisant des valeurs de voxel négatives (APPGML et ADMM-Reg avec pénalité quadratique). Visuellement, les méthodes s'appuyant sur le DIP (DIPRecon et DNA) semblent incorporer les détails de l'image anatomique dans l'image TEP de manière plus naturelle qu'en utilisant la méthode de Boswher avec BSREM, APPGML ou ADMM-Reg dans le cerveau. Pour fournir une initialisation optimale aux algorithmes DNA et DIPRecon, nous avons proposé d'utiliser une image BSREM de départ avec RDP et un paramètre de pénalité favorisant un meilleur contraste mais avec un bruit plus élevé, par rapport à la configuration de BSREM se rapprochant de la routine clinique. Cette image de départ a ensuite été débruitée par le DIP en utilisant la méthode EMV pour arrêter automatiquement le DIP avant le surapprentissage. Cette méthode a permis d'obtenir une bonne première estimation quantitative et qualitative et pourrait être appliquée à d'autres algorithmes de reconstruction utilisant un réseau DIP, en particulier les algorithmes de bout en bout s'appuyant sur les travaux de [47].

Dans le chapitre suivant, nous allons étudier plus en détail l'ensemble des hyperparamètres liés à l'algorithme DNA et leur influence sur le comportement de l'algorithme. Nous partirons du choix de l'initialisation de l'algorithme, puis verrons les hyperparamètres liés à l'algorithme

ADMM-Reg, et finirons par ceux liés à l'entraînement d'un réseau de neurones DIP.

Chapitre 3

Étude et optimisation des hyperparamètres du DNA

3.1 Introduction

L'introduction de nouveaux algorithmes de reconstruction TEP et de nouvelles fonctions de pénalité a permis d'améliorer les images, tant sur le plan qualitatif que quantitatif. L'utilisation d'algorithmes fondés sur l'apprentissage profond, et en particulier d'algorithmes fondés sur le DIP, ajoute également des hyperparamètres liés à l'entraînement du réseau de neurones. L'étendue des images reconstruites possibles a été élargie, mais au prix de l'introduction d'un nombre croissant d'hyperparamètres. Ces hyperparamètres sont généralement réglés expérimentalement, pour un réplikat des données brutes de Poisson, mais sont d'une importance capitale pour la performance des algorithmes de reconstruction.

Dans cette étude, nous nous sommes concentrés sur l'algorithme DNA présenté dans le chapitre précédent, qui utilise l'algorithme ADMM pour découpler la vraisemblance de Poisson de l'apprentissage DIP. L'ensemble des hyperparamètres du DNA se découpe en plusieurs catégories :

- hyperparamètres en lien avec l'initialisation du DNA,
- hyperparamètres en lien avec la contrainte de non-négativité,
- hyperparamètres en lien avec le DIP,
- nombres d'itérations des blocs itératifs du DNA.

Ce chapitre abordera tous les hyperparamètres du DNA. Nous verrons les méthodes utilisées pour ceux qui ont été optimisés de façon automatique. Certains hyperparamètres ont été déterminés parmi une liste finie de valeurs possibles après expérience à travers plusieurs réplicats statistiques. Enfin, nous verrons l'influence de certains hyperparamètres sur la reconstruction afin d'avoir une intuition pour les choisir. Certaines méthodes et intuitions seraient applicables à l'algorithme DIPRecon qui utilise aussi un algorithme ADMM avec une contrainte DIP, et possiblement à d'autres classes d'algorithmes utilisant le DIP (algorithmes de bout en bout par exemple).

3.2 Ensemble des hyperparamètres du DNA

L'ensemble des hyperparamètres de l'algorithme DNA tels qu'utilisés dans le chapitre 2 est présenté dans la figure 3.1. Dans ce chapitre, nous ne traiterons pas de l'image d'entrée du DIP ni du choix du nombre de sauts de connexion dans l'architecture du réseau de neurones, qui seront étudiés dans le chapitre suivant. De plus, nous reviendrons sur le choix de l'algorithme interne optimisant la vraisemblance (ADMM-Reg a été utilisé dans le chapitre 2) dans la partie 3.4.2.

Hyperparamètres liés à l'initialisation
- image de départ - nombre d'itérations du DIP / méthode d'arrêt précoce
Hyperparamètres liés à la contrainte de non-négativité
- ρ_2 : paramètre de pénalité ADMM associé à la contrainte de non-négativité
Hyperparamètres liés à l'entraînement du DIP
- optimiseur du DIP - mise à l'échelle de l'entrée et de l'étiquette du DIP - ρ_1 : paramètre de pénalité ADMM associé à la contrainte du DIP - taux d'apprentissage du DIP - f : architecture du DIP (dont sauts de connexion) - z : entrée du DIP
Nombre d'itérations
- nombre d'itérations du DIP (pendant la reconstruction) - nombre de sous-itérations internes - K : nombre d'itérations internes - N : nombre d'itérations externes

FIGURE 3.1 – Liste des hyperparamètres liés à l'optimisation du DNA tel que présenté dans le chapitre précédent.

3.2.1 Fantômes et simulations analytiques

3.2.1.1 Fantômes

Les résultats présentés dans ce chapitre ont été réalisés pendant toute la durée de la thèse et à différents moments. Ils sont donc présentés dans ce manuscrit par rapport au fantôme utilisé à un moment t pendant la thèse. Pour chaque hyperparamètre, nous préciserons quel fantôme avec quelle simulation ont été utilisés pour mener l'expérience. Le lecteur ne doit pas chercher de justification au fantôme choisi pour chaque expérience, sauf si celle-ci est indiquée explicitement.

Nous parlerons de “fantôme cérébral du chapitre 2” pour faire référence au fantôme présenté dans la figure 2.5 de la partie 2.4.2.1. De même, nous parlerons du “fantôme cylindrique 2” lorsque nous utilisons le fantôme présenté dans la partie 2.7.2.1 du chapitre 2, figure 2.10. Voyons les autres fantômes qui ont été utilisés pendant la thèse et notamment dans l'étude des hyperparamètres du DNA.

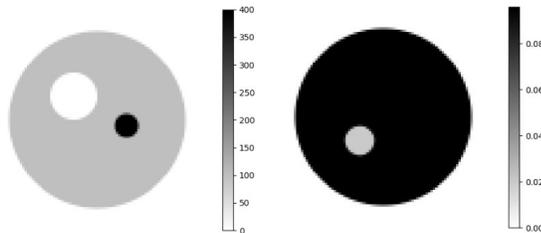


FIGURE 3.2 – Fantôme cylindrique 1. A gauche, la TEP, à droite l'image d'atténuation en cm^{-1} utilisée aussi comme entrée du DIP.

3.2.1.1.1 Fantôme cylindrique 1 Ce fantôme a servi pour effectuer des simulations de type ^{90}Y . La GT TEP et l'image d'atténuation sont présentées dans la figure 3.2. L'image d'atténuation a aussi été utilisée avec ce fantôme en entrée du DIP. La valeur TEP dans la région froide est choisie avec une valeur strictement positive, ici 10. Les valeurs du fond et de la région chaude ont été fixées respectivement à 100 et 400. L'atténuation a été fixée à $0,096\text{cm}^{-1}$

partout sauf une zone sous-atténuée avec une valeur de $0,02\text{cm}^{-1}$.

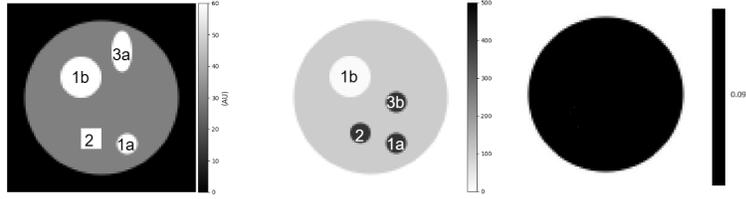


FIGURE 3.3 – Fantôme cylindrique 3. À gauche l'image anatomique utilisée en entrée du DIP, au milieu la GT TEP, à droite l'image d'atténuation en cm^{-1} .

3.2.1.1.2 Fantôme cylindrique 3 Ce fantôme a servi pour effectuer à la fois des simulations de type ^{90}Y et de type FDG. L'image anatomique, l'image TEP et l'image d'atténuation sont présentées dans la figure 3.3. Ce fantôme a été créé avec trois types de régions :

- une région chaude et froide, chacune de forme et de taille identiques entre la TEP et l'image anatomique (régions 1a et 1b),
- une région chaude dont la forme est différente entre la TEP et l'image anatomique (région 2),
- une région chaude n'apparaissant que dans la TEP (région 3a) et une région n'apparaissant que dans l'image anatomique (région 3b).

L'image d'atténuation a été choisie uniforme avec une valeur de $0,096\text{cm}^{-1}$.

Ces deux fantômes sont discrétisés en 112×112 voxels de $4\text{ mm} \times 4\text{ mm}$ et les reconstructions ont été effectuées avec cette même discrétisation.

3.2.1.2 Simulations analytiques

Pendant la thèse, nous avons effectué deux types de simulations servant à étudier l'influence des hyperparamètres de la figure 3.1, toutes deux avec la géométrie du scanner Biograph mMR de Siemens :

- des simulations ^{90}Y , comme décrites dans la partie 2.7.2.1,
- des simulations FDG, comme décrites dans la partie 2.4.2.

3.3 Hyperparamètres liés à l'initialisation utilisée pour le DNA

Les auteurs du DIPRecon [37] ont proposé d'initialiser leur algorithme à partir d'une image pré-reconstruite. Pour cela, une image est reconstruite avec l'algorithme MLEM avec 60 itérations, puis débruitée avec un réseau DIP avec 300 itérations de l'optimiseur L-BFGS. Dans cette thèse, nous avons repris cette idée, en utilisant plutôt l'optimiseur Adam (comme expliqué plus tard, partie 3.5.1.3), et pour un nombre d'itérations déterminé automatiquement par l'algorithme EMV (présenté en détail dans la partie 3.3.2.2.2). L'image de départ choisie sera obtenue avec l'algorithme BSREM+RDP comme expliqué dans la partie 3.3.1.

3.3.1 Choix de l'image de départ

Dans cette partie, nous nous sommes intéressés à l'image de départ qui sera débruitée par DIP pour être utilisée comme initialisation du DNA.

3.3.1.1 Expériences

Nous avons choisi plusieurs images de départ à partir d’images reconstruites avec les algorithmes itératifs suivants :

- MLEM 60 itérations comme dans [37],
- BSREM a été utilisé avec un paramétrage proche de celui employé en routine clinique, avec 30 itérations et 28 sous-ensembles et une pénalité RDP [2]. Le paramètre de pénalité est choisi manuellement pour autoriser un léger bruit dans l’image reconstruite et favoriser un meilleur contraste,
- ADMM-Reg 100 et 300 itérations sans pénalité.

Une architecture DIP sans ReLU (respectivement avec ReLU) a été utilisée pour initialiser ensuite l’algorithme DNA (respectivement le DIPRecon). Nous comparons ces différentes images de départ en étudiant les métriques associées aux images reconstruites définies dans la partie 2.4.2.3. Deux expériences ont été menées à différents moments de la thèse :

- La première a été menée et présentée en poster à la conférence IEEE Medical Imaging Conference 2022 à partir des simulations ^{90}Y du fantôme cylindrique 1. La standardisation a été utilisée pour mettre à l’échelle l’image étiquette et l’image d’entrée du DIP. Le DIPRecon et le DNA ont été utilisés avec une image de départ MLEM, BSREM ou ADMM-Reg 100 itérations. Les autres hyperparamètres ont été réglés manuellement. 300 itérations externes ont été réalisées pour le DNA avec image de départ BSREM. 100 itérations externes ont été réalisées dans les autres cas afin d’obtenir les tendances.
- La deuxième a été réalisée ensuite sur le fantôme cylindrique 2 avec le paramétrage proposé dans la partie 2.7.2.2. MLEM, BSREM et ADMM-Reg 300 itérations ont été utilisés comme image de départ. 300 itérations externes ont été réalisées dans chaque cas.

3.3.1.2 Résultats

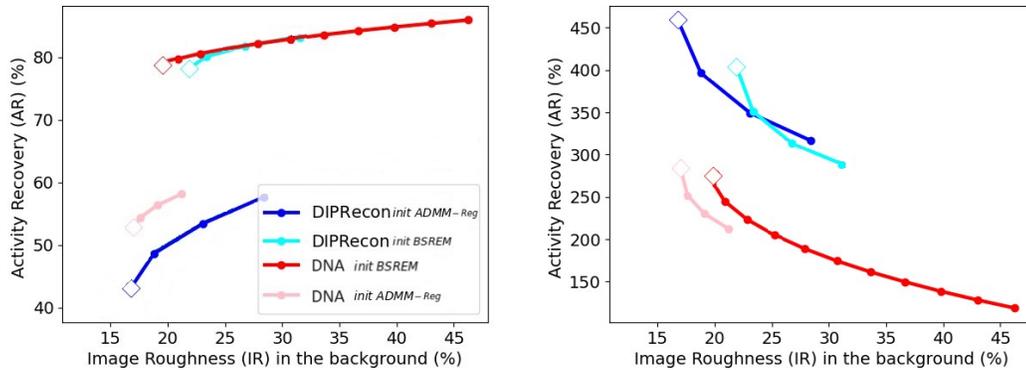


FIGURE 3.4 – Courbes de compromis moyennées sur 100 réplicats sur le fantôme cylindrique 1 dans la ROI de la tumeur (gauche) et la ROI de la zone froide (droite), pour le DNA et le DIPRecon, en faisant varier l’image de départ (ADMM-Reg 100 itérations et BSREM clinique). Les diamants blancs correspondent à l’initialisation DIP-BSREM pour chaque algorithme.

La figure 3.4 représente les métriques obtenues avec le DNA dans le fantôme cylindrique 1, en utilisant différentes images de départ (BSREM et ADMM-Reg 100 itérations). Très clairement dans la zone chaude (gauche), l’image de départ BSREM permet une meilleure initialisation du DNA et du DIPRecon par rapport aux autres images de départ. Dans la zone froide, le DNA est mieux initialisé que le DIPRecon et parvient à une meilleure récupération d’activité. Le DNA avec l’image ADMM-Reg à 100 itérations a une activité comparable à celle l’initialisation avec BSREM, mais avec une rugosité de l’image plus faible.

La figure 3.5 représente les métriques obtenues avec le DNA dans le fantôme cylindrique 2 en utilisant différentes images de départ (MLEM, BSREM et ADMM-Reg 300 itérations).

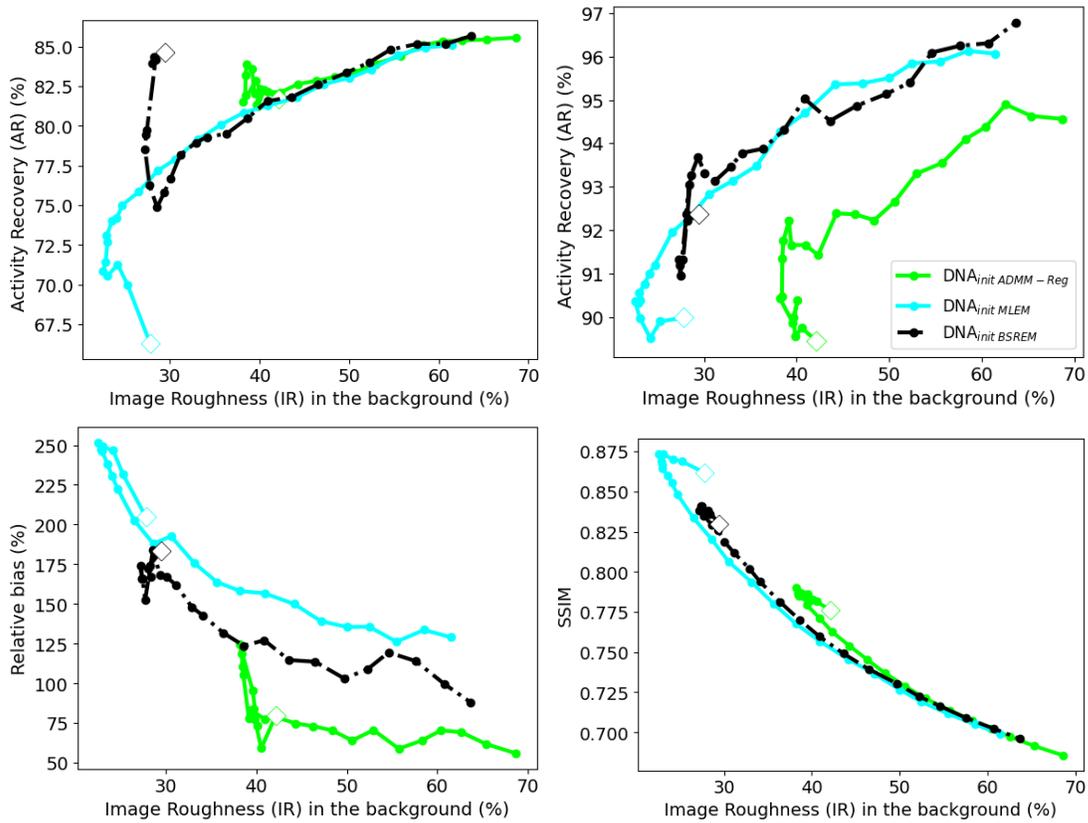


FIGURE 3.5 – Courbes de compromis moyennées sur 20 réplicats sur le fantôme cylindrique 2 dans la ROI de la tumeur “TEP seule” (haut gauche), la ROI de la tumeur “TEP match” (haut droite), la ROI de la zone froide (bas gauche) et le SSIM dans l’ensemble du fantôme (bas droite), pour le DNA et le DIPRecon, en faisant varier l’image de départ (ADMM-Reg 300 itérations, MLEM 60 itérations et BSREM clinique). Les diamants blancs correspondent à l’initialisation DIP-BSREM pour chaque algorithme.

Nous allons souligné ici les différences avec les résultats précédents. On remarque tout d’abord que l’initialisation avec l’image ADMM-Reg à 300 itérations est cette fois plus bruitée que les initialisations avec MLEM et BSREM. En revanche dans la région froide, elle obtient un biais plus faible que les initialisations avec MLEM et BSREM, ce qui permet au DNA d’avoir une meilleure courbe de compromis. Une autre différence concerne le DNA avec image de départ MLEM ou BSREM. En effet, les courbes de compromis se superposent dans la région “TEP match”. De plus, dans la région “TEP seule”, l’initialisation DIP-BSREM est meilleure que l’initialisation DIP-MLEM, mais la courbe pour une image de départ BSREM chute pour finalement se superposer à la courbe pour une image de départ MLEM. Le DNA avec image de départ BSREM présente toutefois un meilleur compromis dans la région froide qu’avec image de départ MLEM, bien que ce dernier paramétrage permette d’obtenir un meilleur SSIM couplé à une faible rugosité de l’image dans les premières itérations.

3.3.1.3 Discussion

Pour choisir l’image de départ du DNA, une idée est de reconstruire une image avec un algorithme ML (MLEM, ADMM-Reg sans régularisation) avec peu d’itérations par rapport à ce qu’il faudrait pour atteindre la convergence des algorithmes. Bien que cela permette de limiter le bruit à retirer par le DIP, cela a pour conséquence des régions qui ne sont pas encore bien convergées, ce qui impacte l’initialisation du DNA, puis ses performances. L’image MLEM 60 itérations utilisée par [37] n’est donc pas toujours une bonne image de départ. En outre, l’image ADMM-Reg à 300 itérations semble présenter une activité plus fidèle à la GT par rapport à l’image obtenue à 100 itérations, mais le débruitage DIP est finalement plus bruité. En définitive, nous avons retenu pour cette thèse l’image de départ BSREM. En effet, le DNA et le DIPRecon ont obtenu les meilleurs résultats avec l’image de départ BSREM pour l’un des fantômes. Le DNA avec image de départ BSREM est moins performant sur l’autre fantôme, mais qui procurent toutefois des résultats meilleurs qu’avec une image de départ MLEM. Ces résultats s’expliquent notamment par les meilleurs compromis obtenus dans les régions chaudes avec BSREM par rapport aux algorithmes MLEM et ADMM-Reg dans le chapitre 2, partie 2.7.3.

Dans tous les résultats obtenus, le DNA a permis une réduction de biais au prix d’une augmentation de la rugosité de l’image, quelle que soit l’image de départ. Le choix de l’image de départ impacte les performances du DNA, bien qu’il soit toujours capable d’effectuer une réduction de biais. Comme déjà décrit dans le chapitre précédent, le DNA peut être vu comme un chemin permettant de partir d’une image pré-reconstruite avec un algorithme itératif et débruitée par DIP, vers une image ADMM-Reg sans régularisation.

3.3.1.4 Conclusion

Une reconstruction avec l’algorithme BSREM+RDP est adaptée en tant qu’image de départ du DNA ou du DIPRecon étant donné la bonne quantification dans les régions chaudes comparée aux initialisations avec MLEM ou ADMM-Reg. Cette image de départ a été choisie pour obtenir les résultats du DNA dans le chapitre 2.

3.3.2 Méthodes d’arrêt précoce pour le DIP

Cette partie a d’abord été étudiée par un stagiaire de l’École Centrale de Nantes dont j’ai participé à l’encadrement, au début de la deuxième année de thèse.

3.3.2.1 Intuition pour les méthodes d’arrêt précoce pour le DIP

Dans une tâche de débruitage, le DIP cherche à optimiser une fonction de coût dont le sens mathématique est de reproduire une image x_{label} (cf. équation (2.2), partie 2.2.2). Un réseau DIP peut être utilisé pour débruiter une image grâce à sa capacité intrinsèque à reproduire plus facilement une image naturelle que du bruit [92]. Ainsi, avant de reproduire l’image x_{label} , le réseau DIP peut la débruiter.

Par la définition même du problème d'optimisation que le DIP cherche à résoudre, un réseau DIP possède par nature un régime de surapprentissage de l'image x_{label} si son nombre de paramètres est suffisamment important. Considérons une mesure de similarité, comme la MSE, le SSIM ou le PSNR (Peak Signal to Noise Ratio) entre la GT et la sortie du réseau DIP calculée à chaque itération (cf. figure 3.6). Les courbes ainsi formées possèdent un extremum correspondant à une image d'intérêt produite par le réseau DIP (cf. partie 2.2.2). Le DIP entre dans son régime de surapprentissage après ce point. Les auteurs dans [99] ont proposé deux méthodes d'arrêt précoce pour déterminer une itération proche de celle de l'extremum désiré. L'idée est de trouver une fonction qui ne dépend pas de la GT et dont l'extremum est atteint à une itération proche de celle de l'extremum d'une des mesures de similarité avec la GT. Ce principe est illustré sur la figure 3.6.

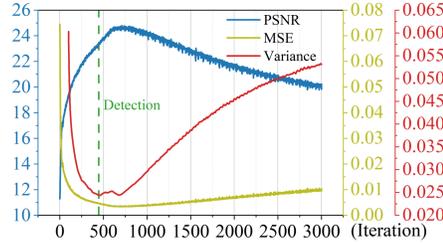


FIGURE 3.6 – Principe de la variance mobile proposé par [99] pour pratiquer un arrêt précoce pour le DIP. La figure est extraite de l'article [99].

Les deux méthodes proposent une fonction calculée à partir de la variance mobile entre les sorties du DIP à différentes itérations. Elles s'appuient sur l'heuristique suivante : le régime du réseau DIP où celui-ci propose la meilleure version débruitée de l'image x_{label} est le moment où les images diffèrent le moins entre itérations successives du DIP.

Voyons maintenant plus en détail les mathématiques permettant de calculer cette variance mobile.

3.3.2.2 Calcul de la variance mobile

3.3.2.2.1 Méthode de la WMV La méthode de la WMV (Window Moving Variance) calcule la variance mobile entre W images de sorties consécutives du réseau DIP. On parle alors de fenêtre de taille W , qui se déplace d'une image de sortie du réseau DIP pour calculer la WMV à chaque itération. Elle est donnée par l'équation suivante :

$$\text{VAR}^t = \frac{1}{W} \sum_{w=0}^{W-1} \|x^{t+w} - \frac{1}{W} \sum_{i=0}^{W-1} x^{t+i}\|_2^2 \quad (3.1)$$

Où t est l'itération courante du DIP. Dans cette équation, les images x sont aplaties en 1D.

Le paramètre W contrôle le nombre d'images que l'on considère en même temps pour calculer la variance mobile. Plus W est petit, plus la courbe de variance mobile va être chaotique. Inversement, plus W est grand, plus la courbe de variance va être lissée.

3.3.2.2.2 Méthode de l'EMV La méthode EMV (Exponential Moving Variance) calcule d'abord une moyenne mobile exponentielle (EMA - Exponential Moving Average) entre les images de sorties du DIP à différentes itérations, pondérées par le facteur de décroissance exponentielle α_{EMV} . Une variance mobile exponentielle est alors calculée à partir de la moyenne mobile exponentielle et de la variance mobile exponentielle à l'itération précédente selon la formule suivante :

$$\text{EMA}^{t+1} = (1 - \alpha_{\text{EMV}})\text{EMA}^t + \alpha_{\text{EMV}}x^{t+1} \quad (3.2)$$

$$\text{EMV}^{t+1} = (1 - \alpha_{\text{EMV}})\text{EMV}^t + \alpha_{\text{EMV}}(1 - \alpha_{\text{EMV}})\|x^{t+1} - \text{EMA}^t\|_2^2 \quad (3.3)$$

avec $EMA^0 = 0$ et $EMV^0 = 0$.

Le paramètre α de l'algorithme EMV a la même influence que le paramètre W de l'algorithme WMV.

3.3.2.2.3 Implémentation numérique Pour ces deux méthodes, le but est de trouver l'itération correspondant au minimum de variance mobile. Ceux-ci sont appliqués pendant l'optimisation du DIP. Les variances mobiles sont donc connues uniquement jusqu'à l'itération du DIP courante. En pratique, on émet l'hypothèse que les courbes de variance mobile remontent de manière globale pendant un certain nombre d'itérations. Le but est d'arrêter l'optimisation du DIP une fois qu'un minimum de la variance mobile est déterminé.

En pratique, un nombre de patience P est utilisé pour pouvoir limiter la quantité d'images nécessaires à calculer par le DIP pour connaître ce minimum. Numériquement, le minimum t_{\min} est accepté lorsque la variance à l'itération t_{\min} n'a pas été amélioré pendant P itérations.

3.3.2.3 Expériences

Dans cette thèse, nous avons préféré l'algorithme EMV à l'algorithme WMV, car celui-ci à l'avantage de ne devoir garder en mémoire qu'une image pour calculer la variance mobile, contrairement à la méthode WMV qui nécessite d'enregistrer W images, avec W la taille de la fenêtre choisie.

Pour tester la méthode EMV pour déterminer une image de sortie du DIP intéressante, nous avons effectué un débruitage DIP. Pour cela, nous avons considéré le fantôme cérébral du chapitre 2, avec simulations FDG. L'image de départ BSREM calculée dans la partie 2.4.2.2 a été utilisée comme image étiquette.

Le DIP et l'EMV ont utilisé le même paramétrage que celui servant à initialiser le DNA dans le chapitre 2 (cf. partie 2.4.2.2). En particulier, pour l'algorithme EMV, $\alpha = 0,1$ avait été proposé par les auteurs de [99]. Cependant, la courbe de variance mobile était trop chaotique pour certains réplicats utilisés dans la partie 2.4.2.2 avec cette valeur. Nous avons donc choisi $\alpha_{EMV} = 0,0251$ qui a permis de déterminer un minimum pour chaque réplicat dans la partie 2.4.2.2. Dans les résultats présentés ensuite, $P = 500$ a été choisi uniquement pour visualiser la tendance globale de la courbe EMV à augmenter après avoir atteint son minimum. $P = 200$ a été utilisé dans la partie 2.4.2.2 et aurait donné les mêmes résultats ici. La courbe obtenue par l'EMV en fonction des itérations pour un réplicat de la partie 2.4.2.2 est étudiée dans cette partie. Elle est comparée à celle de la MSE et du SSIM entre l'image de sortie du réseau DIP et de la GT.

3.3.2.4 Résultats

La figure 3.7 montre l'évolution de la MSE et du SSIM en fonction des itérations du DIP, ainsi que la courbe de variance mobile obtenue avec l'algorithme EMV. La ligne verticale verte correspond à l'itération déterminée par l'algorithme EMV (ici 550 itérations). Le minimum de MSE et le maximum de SSIM ne sont pas faciles à déterminer visuellement étant donné le comportement assez chaotique de ces deux métriques en fonction des itérations. Cependant, nous pouvons repérer une tendance globale des courbes associées à ces métriques. Nous parlerons donc de minimum ou de maximum pour désigner l'extremum qu'un humain choisirait visuellement à partir des courbes. La courbe de la MSE décroît jusqu'à 600 itérations environ puis augmente, comme attendu pour un réseau DIP. De même, la courbe du SSIM croît puis décroît à partir de 800 itérations environ. La courbe obtenue par l'algorithme EMV décroît d'abord très rapidement puis oscille, mais de façon moins chaotique que les deux autres métriques. On remarque tout de même une tendance de la courbe EMV à remonter à partir de la ligne verticale autour de 550 itérations.

La figure 3.8 représente l'image de départ BSREM et les images débruitées par le DIP, pour les itérations correspondant aux extrema des courbes EMV, MSE et SSIM de la figure 3.7. Les images produites par le réseau DIP aux minimums de MSE et d'EMV sont visuellement très similaires, alors que celle correspondant au maximum de SSIM semble être légèrement plus bruitée.

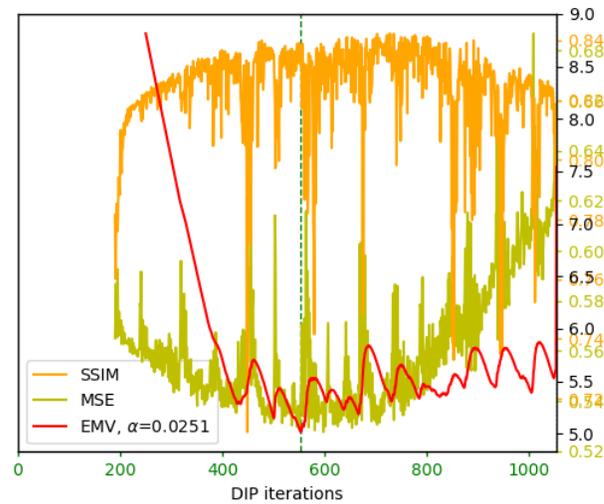


FIGURE 3.7 – MSE, SSIM et EMV (échelle logarithmique) pour un débruitage DIP en fonction des itérations du DIP. Les 200 premières itérations ont été retirées pour une meilleure lisibilité. La ligne verticale verte indique l'itération trouvée par l'algorithme EMV.

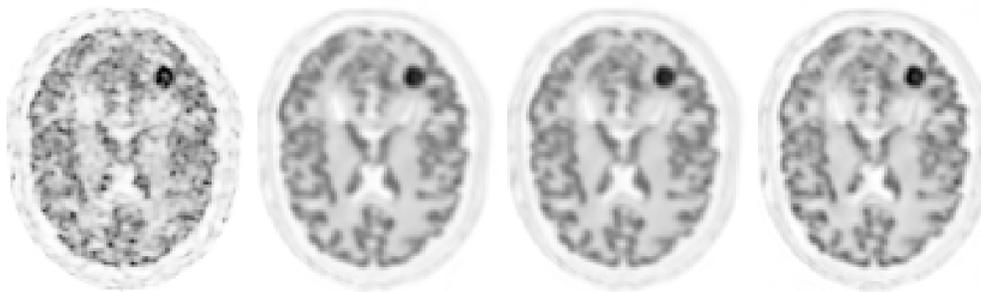


FIGURE 3.8 – Images obtenues après un débruitage DIP de l'image de départ BSREM présentée à gauche. La deuxième image est obtenue à l'itération trouvée par l'algorithme EMV. La troisième correspond à l'image obtenue après 600 itérations du DIP (correspondant environ au minimum de MSE). L'image de droite correspond à l'image obtenue après 800 itérations du DIP (correspondant environ au maximum de SSIM).

3.3.2.5 Discussion

Malgré les oscillations de la courbe obtenue avec la méthode EMV, l'algorithme EMV permet de s'approcher de la zone où la MSE est minimale. L'itération obtenue par l'algorithme EMV (550 itérations du DIP) est en revanche plus éloignée du maximum de SSIM (autour de 800 itérations du DIP). Cependant, la MSE et le SSIM ne sont pas totalement en accord sur l'itération de l'extremum, et l'image correspondant au maximum de SSIM est légèrement plus bruitée que celle obtenue au minimum de MSE. Ainsi, pour ce problème de débruitage, l'algorithme EMV semble adapté car il calcule une itération proche de celle correspondant au minimum de MSE.

3.3.2.6 Conclusion

L'algorithme EMV permet de déterminer une itération proche de celle correspondant au minimum de MSE avec la GT au cours des itérations d'un débruitage DIP. Il permet de choisir une image produite par le réseau DIP de façon automatique pour initialiser le DNA. Il a été utilisé dans le chapitre 2.

3.3.3 Impact de la ReLU finale dans un débruitage avec un réseau DIP

Dans le chapitre 2, nous avons montré que l'architecture avec ReLU finale utilisée pour le DIP pouvait exhiber des artefacts en mettant des zones à zéro. Dans cette partie, nous voulons montrer sur un autre plan que l'architecture avec ReLU finale est moins intéressante que l'architecture sans ReLU finale lors du débruitage DIP initial. Cette partie ne constitue pas le traitement d'un hyperparamètre de la figure 3.1.

3.3.3.1 Expériences

Nous utilisons ici des simulations FDG du fantôme cérébral du chapitre 2. Nous avons étudié des courbes de compromis au cours des itérations du débruitage DIP pour une architecture avec ou sans ReLU, correspondant à ce qui est utilisé pour initialiser le DIPRecon et le DNA. Les paramètres des débruitages sont les mêmes que ceux expliqués dans la partie 2.4.2.2. 2000 itérations du DIP ont été réalisées.

3.3.3.2 Résultats

La figure 3.9 montre les courbes de compromis obtenues pour un débruitage DIP pour une architecture avec ou sans ReLU finale. Pour les deux architectures étudiées, toutes les métriques s'améliorent en même temps que la rugosité de l'image diminue lors des premières itérations du DIP, puis les métriques restent assez stables en même temps que la rugosité de l'image augmente. La différence principale entre les deux architectures se situe au niveau de la rugosité de l'image minimale atteinte, pouvant être bien inférieure pour l'architecture sans ReLU.

La figure 3.10 montre l'évolution de la récupération d'activité dans la région tumorale en fonction des itérations du DIP pour les deux architectures. L'architecture sans ReLU atteint plus rapidement une récupération d'activité stable que l'architecture avec ReLU.

3.3.3.3 Discussion

Cette expérience montre tout d'abord des métriques proche de la GT obtenue en moins d'itérations pour l'architecture sans ReLU par rapport à l'architecture avec ReLU. En outre, des métriques proches de la GT peuvent être obtenues pour une plus faible rugosité de l'image pour l'architecture sans ReLU. Ainsi, cela explique que l'initialisation du DIPRecon dans la partie 2.5.2 soit moins bonne que celle du DNA. De plus, même en choisissant l'itération pour arrêter le DIP manuellement (sans l'algorithme EMV), le DIPRecon ne pourrait pas être mieux initialisé que le DNA.

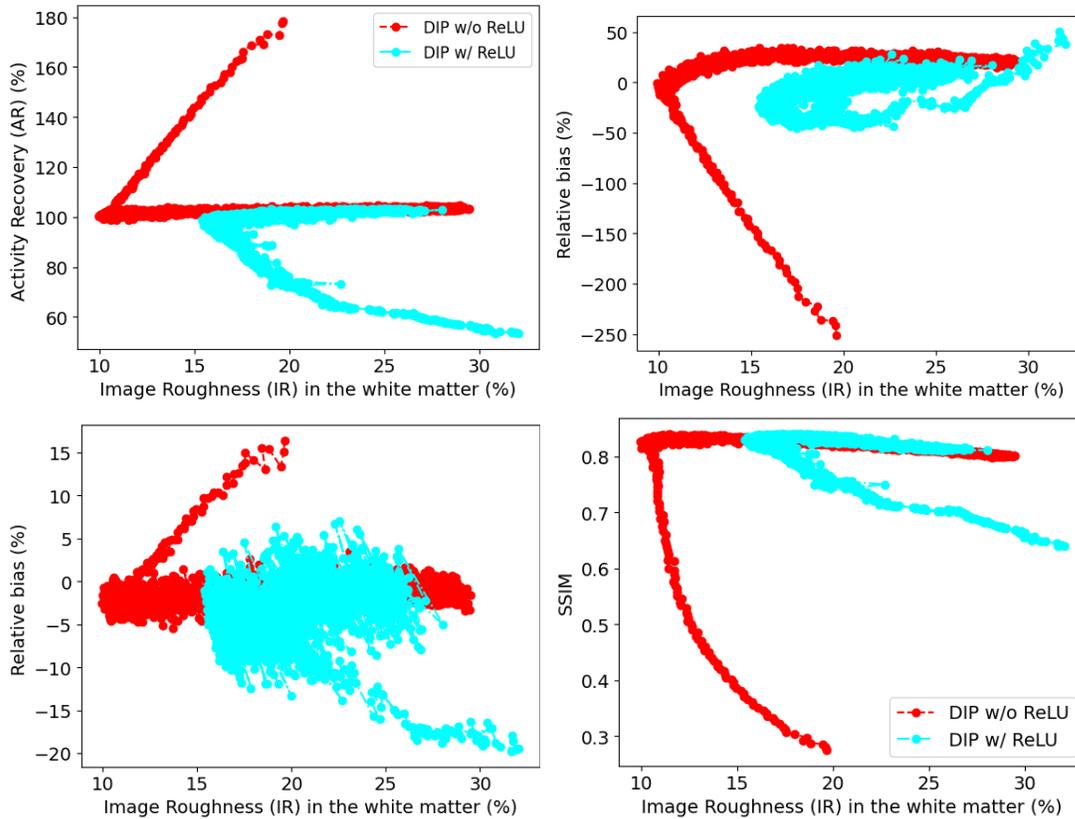


FIGURE 3.9 – Courbes de compromis moyennées sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur (*haut gauche*), la ROI du LCR (*haut droite*), la ROI de la matière blanche (*bas gauche*) et le SSIM dans l'ensemble du cerveau (*bas droite*). Les courbes montrent l'évolution au cours des itérations des métriques sur les initialisations DIP-BSREM utilisées pour le DNA et le DIPRecon. Les 100 premières itérations ont été retirées du graphe pour la lisibilité.

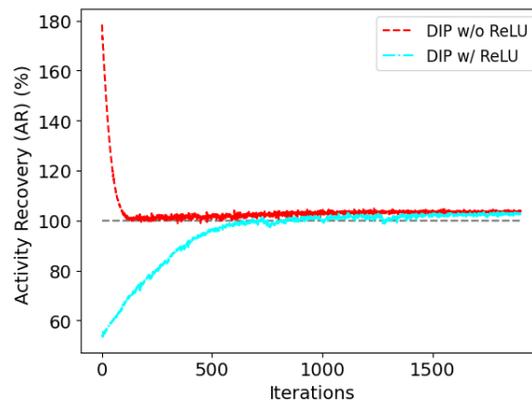


FIGURE 3.10 – Récupération d'activité au cours des itérations du DIP, moyennée sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur pour les initialisations DIP-BSREM utilisées pour le DNA et le DIPRecon. Les 100 premières itérations ont été retirées du graphe pour la lisibilité.

3.3.3.4 Conclusion

Un réseau DIP ayant une architecture sans ReLU finale produit des images avec de bonnes métriques dans les régions d'intérêt et avec un niveau de bruit plus faible que pour une architecture avec ReLU. Ces résultats sont visibles dans le chapitre 2 dans la comparaison entre le DNA et le DIPRecon.

3.4 Hyperparamètres liés à la contrainte de non-négativité

3.4.1 Équilibre des résidus relatifs pour la mise à jour automatique du paramètre ρ_2

Cette partie a d'abord été étudiée par un stagiaire de l'Ecole Centrale de Nantes dont j'ai participé à l'encadrement, au début de la deuxième année de thèse.

3.4.1.1 Méthodes

Pour dériver leur méthode à partir de l'algorithme ADMM, les auteurs dans [59] ont transformé le problème initial (1.49) en un problème d'optimisation avec contrainte d'égalité :

$$\begin{cases} \arg \max_{x,v} y^T \ln(v+r) - 1^T(v+r) - \frac{\rho_2}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2 + g(v+r) \\ \text{avec } Ax = v \end{cases} \quad (3.4)$$

Lors de l'utilisation de l'algorithme ADMM, un paramètre de pénalité ρ_2 doit être défini pour contrôler la vitesse de convergence de l'algorithme.

Tout d'abord, la méthode d'équilibre des résidus a été développée (residual balancing en anglais) [12] pour adapter le paramètre de pénalité ρ_2 . Les résidus sont définis à partir des conditions de faisabilité primale et duale issue du problème d'optimisation. $r^{k+1} = Ax^{k+1} - v^{k+1}$ est le résidu primal à l'itération $k+1$, et $s^{k+1} = -A^T(v^{k+1} - v^k)$ est le résidu dual à l'itération $k+1$ (cf. Annexe B.1.5). Cette méthode est une approche heuristique pour adapter à chaque itération la valeur du paramètre de pénalité ρ_2 en fonction des résidus. L'idée est que les deux résidus convergent à la même vitesse vers zéro (i.e. leur ratio reste proche de 1), sachant que la norme du résidu primal augmente si ρ_2 diminue et que la norme du résidu dual augmente si ρ_2 augmente. À l'itération $k+1$, le paramètre de pénalité ρ_2^{k+1} est mis à jour selon la politique suivante :

$$\rho_2^{k+1} := \begin{cases} \tau \rho_2^k & \text{if } \|r^k\|_2 > \mu \|s^k\|_2 \\ \rho_2^k / \tau & \text{if } \|s^k\|_2 > \mu \|r^k\|_2 \\ \rho_2^k & \text{otherwise} \end{cases} \quad (3.5)$$

μ est un facteur pour comparer le rapport des normes des résidus primal et dual. Si ce ratio est trop grand ou trop faible (par rapport à μ ou $\frac{1}{\mu}$), alors l'intensité de la pénalité doit être modifiée. Si la norme du résidu primal est plus grande que la norme du résidu dual, il faut augmenter la valeur de ρ_2 en la multipliant par $\tau > 1$. Dans le cas inverse, on divise par τ .

La convergence théorique de l'algorithme ADMM avec la politique adaptative (3.5) est assurée si le paramètre de pénalité ρ_2 devient fixe à partir d'une certaine itération.

Le défaut majeur de cette méthode est qu'elle ne prend pas en compte l'échelle du problème ADMM. Cela veut dire qu'un changement de variable $x \leftarrow 10x$ sera vu différemment par les résidus, et entraînant une séquence itérative des variables (x^k, v^k, u^k) différente en fonction de l'échelle utilisée, bien que l'optimum théorique (x^*, v^*, u^*) soit le même (au facteur 10 près) [104] [106].

Pour ne plus dépendre de l'échelle du problème d'optimisation, la méthode d'équilibre des résidus relatifs a été développée par [104] et exploitée lors de cette thèse.

L'idée de l'équilibre des résidus relatifs est la même que l'équilibre des résidus, mais en utilisant des résidus relatifs. Les auteurs de [104] ont défini des résidus relatifs qui sont invariants à une mise à l'échelle du problème d'optimisation (3.4). Le résidu primal relatif est défini par :

$$r_{rel}^{k+1} = \frac{Ax^{k+1} - v^{k+1}}{\max(\|Ax^{k+1}\|_2, \|v^{k+1}\|_2)}, \quad (3.6)$$

et le résidu dual relatif par :

$$s_{rel}^{k+1} = -A^T \frac{(v^{k+1} - v^k)}{\|A^T v^{k+1}\|_2}. \quad (3.7)$$

La politique de mise à jour du paramètre de pénalité ρ_2 avec les résidus relatifs est alors donnée par [104] :

$$\rho_2^{k+1} := \begin{cases} \tau \rho_2^k & \text{if } \|r_{rel}^k\|_2 > \xi \mu \|s_{rel}^k\|_2, \\ \rho_2^k / \tau & \text{if } \|s_{rel}^k\|_2 > \frac{1}{\xi} \mu \|r_{rel}^k\|_2, \\ \rho_2^k & \text{otherwise.} \end{cases} \quad (3.8)$$

L'ajout des résidus relatifs entraîne une invariance du problème d'optimisation (3.4) à une mise à l'échelle, mais au prix de conserver un facteur ξ dans (3.8).

De la même façon que pour la méthode des résidus, la convergence théorique de l'algorithme ADMM avec la méthode des résidus relatifs(3.8) est assurée si le paramètre de pénalité ρ_2 devient fixe à partir d'une certaine itération.

Un autre ajout proposé par [104] est la possibilité de mettre à jour le paramètre multiplicatif τ de façon adaptative, en fonction de l'écart entre les résidus. La politique de mise à jour du paramètre de pénalité ρ_2 est la même que (3.8) mais en utilisant τ^k . Voici la règle utilisée pour mettre à jour le paramètre multiplicatif τ :

$$\tau^{k+1} := \begin{cases} \sqrt{\left(\frac{1}{\xi} \|r_{rel}^{k+1}\|_2 / \|s_{rel}^{k+1}\|_2\right)} & \\ \text{if } 1 \leq \sqrt{\left(\frac{1}{\xi} \|r_{rel}^{k+1}\|_2 / \|s_{rel}^{k+1}\|_2\right)} < \tau^{\max} & \\ \sqrt{\left(\xi \|s_{rel}^{k+1}\|_2 / \|r_{rel}^{k+1}\|_2\right)} & \\ \text{if } \frac{1}{\tau^{\max}} < \sqrt{\left(\frac{1}{\xi} \|r_{rel}^{k+1}\|_2 / \|s_{rel}^{k+1}\|_2\right)} < 1 & \\ \tau^{\max} & \\ \text{otherwise} & \end{cases} \quad (3.9)$$

Si le résidu primal est plus grand que le résidu dual, il faut augmenter le paramètre de pénalité ρ_2 d'après l'heuristique (3.8). Plus les résidus sont éloignés, plus il faut augmenter le paramètre de pénalité ρ_2 , d'où le terme $\sqrt{\left(\frac{1}{\xi} \|r_{rel}^{k+1}\|_2 / \|s_{rel}^{k+1}\|_2\right)}$. Et inversement si le résidu dual est plus grand que le résidu primal. Les auteurs dans [104] décident aussi de borner le paramètre multiplicatif τ par une valeur τ^{\max} pour maintenir la preuve de convergence.

La méthode des résidus relatifs a pour but de choisir la valeur du paramètre de pénalité ADMM ρ_2 de façon automatique à chaque itération. Cependant, elle introduit certes de nouveaux hyperparamètres τ (τ^{\max} si (3.9) est utilisée), μ et ξ , mais dont le réglage est plus intuitif. Dans le chapitre 2, τ^{\max} et μ ont été fixés à 100 et 2 respectivement (cf. partie 2.4.2.2), pour permettre un changement rapide de ρ_2 si son initialisation était loin des valeurs optimales. Quant au paramètre ξ , [104] estime qu'il est peu sensible à régler, et que $\xi = 1$ est un choix qui fonctionne dans plusieurs applications.

Remarque : les équations des résidus et les politiques de mise à jour des paramètres ρ_2 et τ présentées dans cette partie sont les mêmes pour les algorithmes ADMM-Reg et DNA. En effet, les résidus ne dépendent pas de la fonction de coût à optimiser, mais seulement de la contrainte $Ax = v$, commune pour dériver ADMM-Reg et DNA.

3.4.1.2 Conclusion

La méthode des résidus relatifs proposée par [104] a été utilisée dans ce travail afin d'automatiser le choix du paramètre de pénalité ρ_2 . Ses hyperparamètres propres sont plus faciles à régler que le paramètre de pénalité ρ_2 .

3.4.2 Remarques sur le choix de l'algorithme interne optimisant la vraisemblance

Cette partie ne constitue pas le traitement d'un hyperparamètre de la figure 3.1. Nous proposons ici de revenir sur le choix de l'algorithme permettant de résoudre l'équation (2.15a) intégrant le terme d'attache aux données dans le DNA.

Pour cela, dans un premier temps, comparons les métriques des images reconstruites par l'algorithme APPGML par rapport à ADMM-Reg. Ensuite, nous verrons les résultats obtenus en introduisant l'algorithme APPGML de façon empirique à la place de ADMM-Reg au sein du DNA.

3.4.2.1 Comparaison de ADMM-Reg et APPGML

3.4.2.1.1 Contexte Plusieurs algorithmes de reconstruction autorisant les valeurs négatives des voxels ont été développés pour la reconstruction TEP, initialement dans le but de réduire le biais positif dans les régions froides. Une première idée était de décaler les valeurs des voxels de l'image ainsi que des données acquises (ABML [13], APPGML [62], AML [93]). Une autre technique a été proposée en changeant la distribution de Poisson en une distribution gaussienne pour les voxels proches de zéro (NEGML [66; 93]). Les auteurs de [62] ont réalisé une analyse comparative multiobjective entre les algorithmes AML [93], NEGML [93] et APPGML [62] dans le cas de la radioembolisation à l' ^{90}Y . Leurs résultats ont montré que l'algorithme APPGML combiné à une pénalité MRF quadratique est le plus adapté à des fins de visualisation ou de quantification, comparé à AML et NEGML avec post filtrage gaussien. De manière quasi-concomitante, l'algorithme ADMM-Reg a été proposé par [59] aussi dans le cas de la radioembolisation à l' ^{90}Y et l'algorithme HypoC-PML a été développé dans le cadre de basses statistiques [10]. Ces méthodes ont transféré la contrainte de non-négativité de l'espace image vers l'espace des projections, ce qui a l'avantage mathématique de préserver l'hypothèse de la loi de Poisson. Ces deux algorithmes ont conduit à la même réduction de biais, avec l'algorithme HypoC-PML qui converge en moins d'itérations qu'ADMM-Reg [10].

Ainsi, nous avons souhaité comparer les algorithmes APPGML et ADMM-Reg dans un même cadre. HypoC-PML aurait pu aussi être un bon candidat, mais n'avait pas été considéré au début du projet.

3.4.2.1.2 Expériences Nous utilisons ici des simulations ^{90}Y sur le fantôme cylindrique 2. APPGML et ADMM-Reg sont utilisés avec les mêmes paramètres de reconstruction que dans le chapitre 2 (cf. partie 2.4.2.2). Nous faisons aussi varier la borne A de l'algorithme APPGML pour des valeurs de -10, -100 et -1000.

3.4.2.1.3 Résultats Les résultats sont présentés en figure 3.11. Toutes les courbes de compromis sont relativement similaires dans les deux régions chaudes, avec une récupération d'activité s'approchant des 100% pour un niveau très élevé de rugosité de l'image. Le SSIM est aussi similaire pour tous les algorithmes. En revanche, APPGML avec une borne de -10 (valeur absolue plus faible que le fond du fantôme) ne permet jamais une réduction de biais complète dans la région froide. En prenant une borne plus éloignée $A = -100$ (valeur absolue égale au fond du fantôme), le biais s'approche un peu plus de zéro. C'est pour une borne $A = -1000$

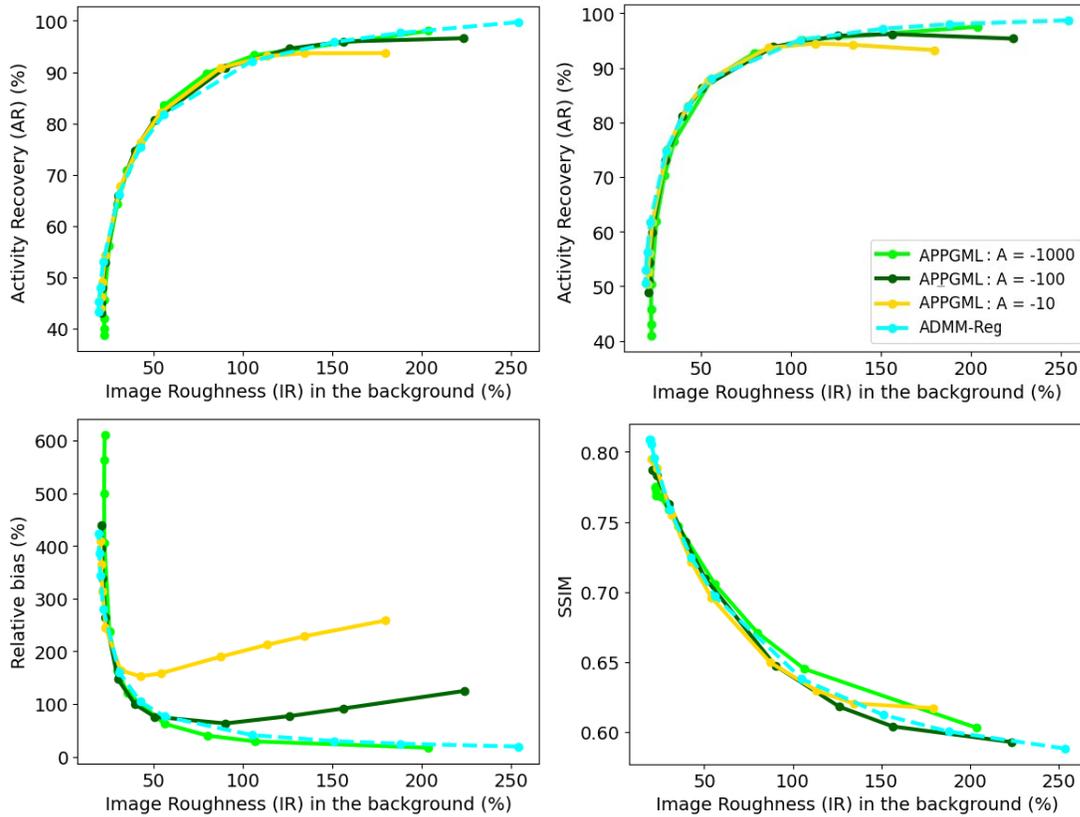


FIGURE 3.11 – Courbes de compromis moyennées sur 20 réplicats pour le fantôme cylindrique 2 en situation ^{90}Y , dans la ROI de la tumeur “TEP seule” (*haut gauche*), la ROI de la tumeur “TEP match” (*haut droite*), la ROI de la zone froide (*bas gauche*) et le SSIM dans l’ensemble du fantôme (*bas droite*), pour les algorithmes APPGML et ADMM-Reg avec pénalité MRF quadratique. Plusieurs valeurs de bornes A sont présentées pour APPGML.

(valeur absolue plus forte que le fond du fantôme) que la réduction de biais est la plus forte, avec un biais alors très proche de zéro. Ce choix de borne conduit à un comportement quasi identique entre APPGML et ADMM-Reg.

3.4.2.1.4 Discussion Les résultats présentés montrent un comportement très proche pour APPGML et ADMM-Reg, pour une borne A bien choisie (au moins égale en valeur absolue au fond du fantôme). Cette comparaison est un ajout à l'étude menée dans [62] qui ne prenait pas en compte ADMM-Reg dans les comparaisons.

Remarques :

- les résultats en FDG sont similaires à ceux présentés dans la figure 3.11 (non présentés ici),
- le choix d'une borne encore plus tolérante pour APPGML ($A < -1000$) ne modifierait pas la réduction de biais de l'algorithme, et n'aurait comme conséquence que l'augmentation du bruit dans les images reconstruites.

3.4.2.2 Remplacement de l'algorithme ADMM-Reg par APPGML à l'intérieur du DNA

Dans la partie 2.1, nous avons justifié le choix d'ADMM-Reg pour autoriser les voxels négatifs tout en préservant la nature Poissonienne des données TEP. Dans cette partie, nous décidons d'élargir les hyperparamètres du DNA au choix d'une méthode de reconstruction itérative autorisant les voxels négatifs.

3.4.2.3 Méthodes

Nous pouvons repartir du même problème d'optimisation de départ que le DIPRecon :

$$\begin{cases} \min_x -L(y|x), \\ \text{subject to } x = f(\theta|z). \end{cases} \quad (3.10)$$

De la même façon que dans la partie 2.2.3, un algorithme ADMM peut être dérivé :

$$x^{n+1} = \arg \min_x -L(y|x) + \frac{\rho_1}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2, \quad (3.11a)$$

$$\theta^{n+1} = \arg \min_\theta \|x^{n+1} - f(\theta|z) + \mu^n\|_2^2 \quad (3.11b)$$

$$\mu^{n+1} = \mu^n + x^{n+1} - f(\theta^{n+1}|z), \quad (3.11c)$$

Considérons l'algorithme APPGML pour résoudre 3.11a. Celui-ci ne résout pas, au sens mathématique, le problème d'optimisation (3.11a). En effet, cet algorithme résout un problème d'optimisation avec une image et des données décalées (cf. partie 1.2.5.6.3). Ainsi, l'hypothèse des données TEP Poisson n'est plus exploitée. L'utilisation d'APPGML pour résoudre (3.11a) est fondé sur une intuition plutôt que sur un développement mathématique. En effet, l'étude comparative dans [62] a montré qu'APPGML pouvait être un bon candidat pour la réduction de biais. Nous avons aussi observé qu'APPGML avait des propriétés quantitatives similaires à celles d'ADMM-Reg pour le fantôme cylindrique 2 (cf. partie 3.4.2.1). Par ailleurs, APPGML avec une borne A très faible se rapproche d'un algorithme de moindres carrés, correspondant à une hypothèse de bruit gaussien sur les données [62; 93]. Cependant, aucune hypothèse sur le type de bruit n'est nécessaire pour le DIP. Nous appellerons DNA-APPGML l'algorithme résolvant l'ensemble d'équations (3.11) en utilisant APPGML. Par analogie au DNA (cf. partie 2.3.2), le nombre d'itérations internes est défini comme le nombre d'itérations utilisé pour résoudre l'équation (3.11a).

3.4.2.3.1 Expériences Ces expériences ont pour but de comparer les algorithmes DNA et DNA-APPGML.

Tout d’abord, nous avons effectué des simulations ^{90}Y du fantôme cylindrique 3. Les paramètres du DNA et du DNA-APPGML ont été fixés comme dans la partie 2.7.2.2, en faisant varier en plus le nombre d’itérations internes des deux algorithmes (respectivement 10, 30 et 4, 14, 28 itérations internes). Une borne $A = -1000$ a été utilisée, correspondant aux meilleurs résultats dans la partie 3.4.2.1. 100 itérations externes ont été réalisées dans chaque cas.

Nous avons aussi effectué des reconstructions sur le fantôme cérébral FDG du chapitre 2. Tous les paramètres pour le DNA et le DNA-APPGML ont été fixés de la même manière que ceux du DNA dans la partie 2.4.2.2. Une borne $A = -10$ a été utilisée. 700 itérations externes ont été réalisées pour le DNA et 300 pour DNA-APPGML.

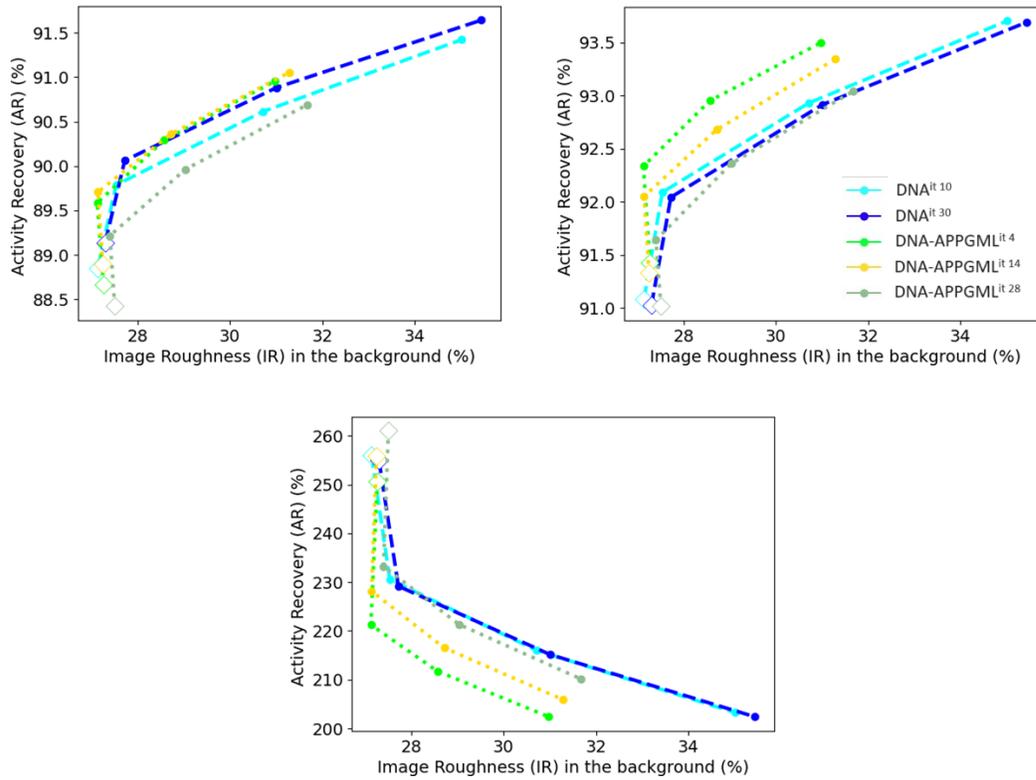


FIGURE 3.12 – Courbes de compromis moyennées sur 20 réplicats pour le fantôme cylindrique 3 en situation ^{90}Y , dans la ROI de la tumeur “TEP seule” (*haut gauche*), la ROI de la tumeur “TEP match” (*haut droite*), la ROI de la zone froide (*bas*), pour l’algorithme DNA proposé et la version modifiée DNA-APPGML. Le nombre d’itérations internes des deux algorithmes a été varié. Les diamants blancs correspondent à la première itération externe pour tous les algorithmes étudiés.

3.4.2.3.2 Résultats Tout d’abord, la figure 3.12 présente les courbes de compromis obtenues avec le fantôme cylindrique 3 en situation ^{90}Y après reconstruction par DNA et DNA-APPGML pour différents nombres d’itérations internes. Les courbes dans la région “TEP seule” (en haut à gauche de la figure 3.12) sont assez comparables. Dans la région “TEP match” (en haut à droite de la figure 3.12), les courbes du DNA-APPGML sont légèrement meilleures que celles du DNA, en particulier pour un faible nombre d’itérations internes. Nous observons le même comportement pour dans la région froide.

Ensuite, la figure 3.13 présente les courbes de compromis obtenues avec le fantôme cérébral FDG du chapitre 2 après reconstruction par DNA et DNA-APPGML. Les courbes de compromis

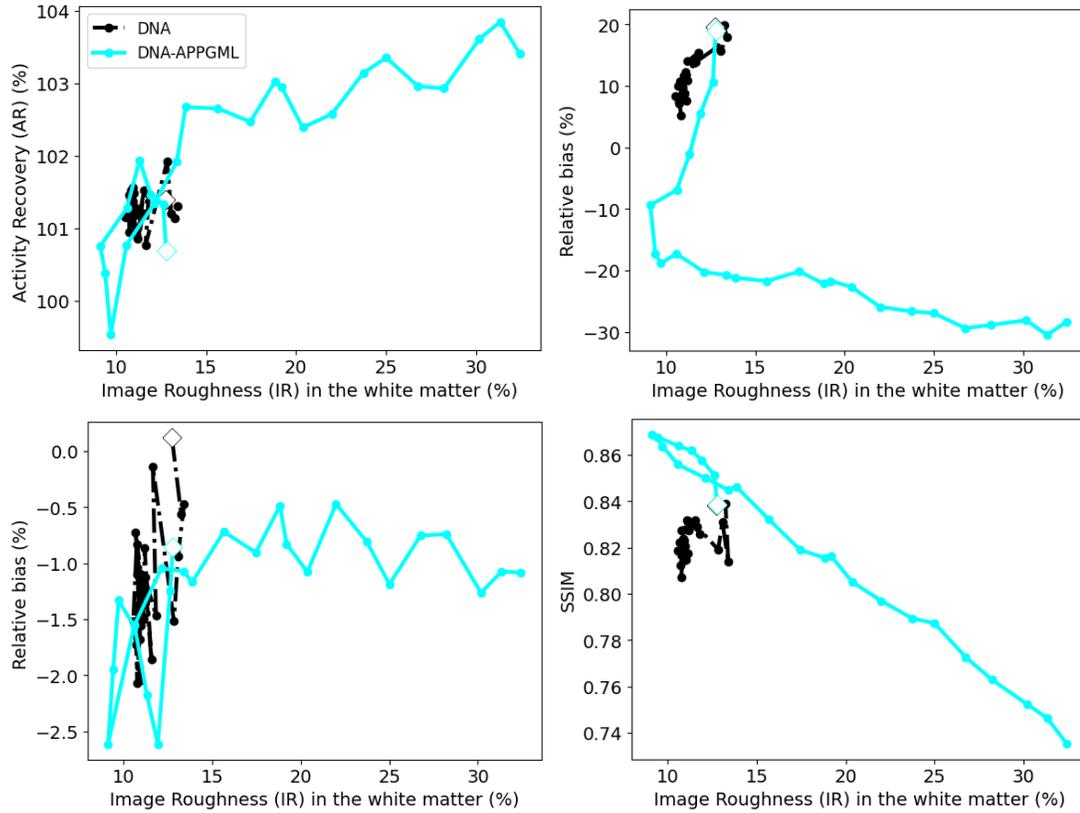


FIGURE 3.13 – Courbes de compromis moyennées sur 20 réplicats pour le fantôme cérébral FDG dans la ROI de la tumeur (*haut gauche*), la ROI du LCR (*haut droite*), la ROI de la matière blanche (*bas gauche*) et le SSIM dans l'ensemble du cerveau (*bas droite*), pour l'algorithme DNA proposé et la version modifiée DNA-APPGML. Le nombre d'itérations internes des deux algorithmes a été varié. Les diamants blancs correspondent à la première itération externe pour tous les algorithmes étudiés.

des deux algorithmes sont assez chaotiques. Pour des valeurs de rugosité de l'image similaires, la récupération d'activité tumorale et le biais relatif dans la matière blanche sont assez proches pour les deux algorithmes. Le DNA-APPGML semble atteindre un SSIM plus élevé et un biais relatif dans le LCR quasi nul pour une rugosité de l'image légèrement plus faible que le DNA. Globalement, le DNA-APPGML introduit du bruit dans la reconstruction de façon plus rapide par rapport au DNA utilisant ADMM-Reg pour les mêmes paramètres de reconstruction.

3.4.2.3.3 Discussion Nous notons tout d'abord la différence dans la vitesse d'évolution du DNA et du DNA-APPGML en fonction des simulations. Dans le cas ^{90}Y , les deux algorithmes ont un comportement similaire, avec DNA-APPGML légèrement meilleur que le DNA dans la région "TEP match" et la région froide. Néanmoins, dans le cas FDG, le DNA prend beaucoup plus d'itérations à faire évoluer les métriques par rapport au DNA-APPGML. Ce dernier peut également reconstruire des images avec des métriques légèrement plus performantes que le DNA.



FIGURE 3.14 – Images obtenues pendant la reconstruction du fantôme cylindrique 3 avec l’algorithme DNA-APPGML et une borne $A = -10$. La première ligne montre les images au bout de 2 et 3 itérations externes du DNA-APPGML. La deuxième ligne montre les images obtenues pendant la troisième itération externe lors de la reconstruction par APPGML au sein du DNA-APPGML (à 5 et 6 itérations internes).

En outre, le DNA-APPGML requiert l’ajustement d’un nouvel hyperparamètre par rapport au DNA, qui est la borne A de l’algorithme APPGML (cf. partie 1.2.5.6.3). L’algorithme DNA-APPGML peut diverger dans le cas d’une borne A choisie trop faible en valeur absolue (proche de zéro). En effet, celui-ci emploie l’algorithme APPGML pour résoudre l’équation (3.11a). Or cette étape demande de reconstruire une image x ayant une borne inférieure A , en optimisant la vraisemblance et une fonction quadratique de cible $f(\theta|z) - \mu$. Cependant, cette image cible peut posséder des valeurs plus faibles que la borne A choisie au préalable. Ce problème est mis en évidence dans la figure 3.14, où les valeurs des pixels de l’image $f(\theta|z) - \mu$ sont inférieurs à la borne A (3.14 bas gauche). Sans troncature pour ramener ces valeurs à la borne A , l’algorithme APPGML est initialisé avec des pixels inférieurs à la borne A , ce qui le fait diverger, ainsi que le DNA-APPGML.

3.4.2.3.4 Conclusion En résumé, voici les avantages et inconvénients du DNA par rapport au DNA-APPGML.

	DNA	DNA-APPGML
Avantages	Dérivation mathématique	Plus rapide pour améliorer les métriques en FDG + métriques légèrement meilleures
Inconvénients	Trop d'itérations pour améliorer les métriques en FDG	Hyperparamètre : Choix du A
Inconvénients	Divergence si ρ_1 trop faible ou nombre d'itérations internes trop élevé	Problème intrinsèque de dérivation : divergence si A trop faible

TABLE 3.1 – Avantages et inconvénients du DNA par rapport au DNA-APPGML.

Nous avons continué à travailler avec l'algorithme DNA utilisant ADMM-Reg dans ce chapitre pour sa justification mathématique ainsi que pour éviter le choix d'une borne A avec le DNA-APPGML.

3.5 Hyperparamètres liés au DIP au sein de la reconstruction

3.5.1 Étude de l'optimiseur du DIP

3.5.1.1 Optimiseurs utilisés en TEP avec un réseau DIP

La fonction de coût du DIP est donnée par l'équation 2.2 du chapitre 2. Elle ne prend en compte qu'une seule paire d'images en comparaison à un entraînement supervisé. L'utilisation d'un algorithme stochastique qui calcule le gradient sur une partie des images de la base de données ne présente donc pas d'intérêt par rapport à un algorithme déterministe pour l'optimisation du DIP.

Dans ce travail, nous avons étudié deux algorithmes d'optimisation utilisés en TEP avec un réseau DIP : Adam et L-BFGS. D'une part, Adam a été utilisé dans les études suivantes : [17; 24; 53; 80; 81; 107]. En particulier, dans [88], Adam a été préféré par rapport à L-BFGS dont les métriques quantitatives oscillaient plus qu'avec Adam. D'autre part, L-BFGS est un algorithme d'optimisation préféré à Adam dans certaines études, par exemple [37; 38; 67; 69] pour ses propriétés de monotonie et sa rapidité de convergence.

Dans les paragraphes suivants, nous allons voir les équations de mise à jour des poids du réseau DIP avec ces algorithmes.

3.5.1.1.1 Adam Adam [56] est un algorithme d'optimisation développé pour l'entraînement d'un réseau de neurones supervisé. Celui-ci est un algorithme d'optimisation d'ordre 1, c'est-à-dire qu'il prend en compte le gradient de la fonction de coût pour calculer les poids du réseau de neurones. Il s'appuie sur l'algorithme de descente de gradient, mais y ajoute deux contributions :

- le vecteur de descente est le premier moment du gradient plutôt que le gradient lui-même, auquel une correction de biais est effectuée. Le premier moment du gradient m^{t+1} à l'itération $t + 1$ est défini par la formule suivante :

$$m^{t+1} = \beta_1 m^t + (1 - \beta_1) \nabla f(\theta^t | z), \quad (3.12)$$

où $\beta_1 \in [0; 1]$. Adam modifie le moment m^{t+1} en effectuant une correction de biais pour obtenir le moment corrigé \hat{m}^{t+1} selon l'expression suivante :

$$\hat{m}^{t+1} = \frac{m^{t+1}}{1 - \beta_1^t}. \quad (3.13)$$

- le taux d'apprentissage est rendu dépendant de chaque poids du réseau, pondéré par le deuxième moment du gradient auquel on a aussi effectué une correction de biais. Voici les formules du deuxième moment v^{t+1} à l'itération $t + 1$, et de sa version corrigée \hat{v}^{t+1} :

$$v^{t+1} = \beta_2 v^t + (1 - \beta_2) \nabla f(\theta^t | z)^2, \quad (3.14)$$

$$\hat{v}^{t+1} = \frac{v^{t+1}}{1 - \beta_2^t}. \quad (3.15)$$

En considérant une division terme à terme des vecteurs, la mise à jour des poids du réseau avec l'optimiseur Adam à l'itération $t + 1$ s'écrit de la façon suivante :

$$\theta^{t+1} = \theta^t - \gamma \frac{\hat{m}^{t+1}}{\sqrt{\hat{v}^{t+1} + \epsilon}}, \quad (3.16)$$

avec γ le taux d'apprentissage, et ϵ une constante pour éviter les problèmes numériques, choisie par défaut à 10^{-8} .

3.5.1.1.2 L-BFGS L-BFGS [108] est un algorithme d'ordre 2, c'est-à-dire qu'il prend en compte le gradient et la matrice hessienne H^t de la fonction de coût pour calculer les poids du réseau de neurones. C'est un algorithme de descente quasi Newtonien, signifiant que la direction de descente $d^t = (H^t)^{-1} \nabla f(\theta^t | z)$ est calculée en pratique avec une approximation B^t de la matrice hessienne.

La mise à jour des poids est donnée par l'équation suivante :

$$\theta^{t+1} = \theta^t - \gamma d^t, \quad (3.17)$$

Cette méthode évite le stockage en machine de la matrice hessienne qui est calculée seulement à partir de P couples $\{(y^t, s^t), \dots, (y^{t-P+1}, s^{t-P+1})\}$ enregistrés à chaque itération t , avec $s^t = \theta^{t+1} - \theta^t$ et $y^t = \nabla f(\theta^{t+1} | z) - \nabla f(\theta^t | z)$. Pour chaque itération t , l'algorithme réalise C itérations internes pour approximer la matrice hessienne au point courant. La matrice B^{t+1} est initialisée avec une matrice B^0 à chaque itération interne, et est calculée de façon récursive pendant les C itérations internes selon la formule suivante :

$$(B^{t+1})^{-1} = (V^t)^T (B^t)^{-1} (V^t) + \rho_t (s^t)^T s^t, \quad (3.18)$$

avec $V^t = I - \rho_t y^t (s^t)^T$, $\rho_t = \frac{1}{(y^t)^T s^t}$.

Après les C itérations internes, l'algorithme L-BFGS a effectué une itération et propose une mise à jour des poids du réseau et donc de l'image de sortie du DIP.

3.5.1.2 Expériences

Nous avons utilisé les simulations FDG du fantôme cérébral du chapitre 2 et avons reconstruit des images légèrement bruitées avec BSREM + RDP (mêmes images que l'image de départ BSREM dans la partie 2.4.2.2 du chapitre 2). Des débruitages ont été réalisés avec un réseau DIP tel qu'utilisé pour initialiser le DNA. Nous avons utilisé deux optimiseurs différents : Adam et L-BFGS. Les paramètres d'Adam sont les mêmes que ceux présentés dans le chapitre 2, partie 2.4.2.2. Le nombre d'itérations internes à L-BFGS a été fixée à 10 comme dans [37], ainsi que sa mémoire a été choisie à $P = 10$. Trois valeurs de taux d'apprentissage ont été testées : 0.01, 0.1 et 1 (valeur par défaut dans Pytorch). 2000 itérations du DIP ont été réalisées avec Adam, et 1000 avec L-BFGS.

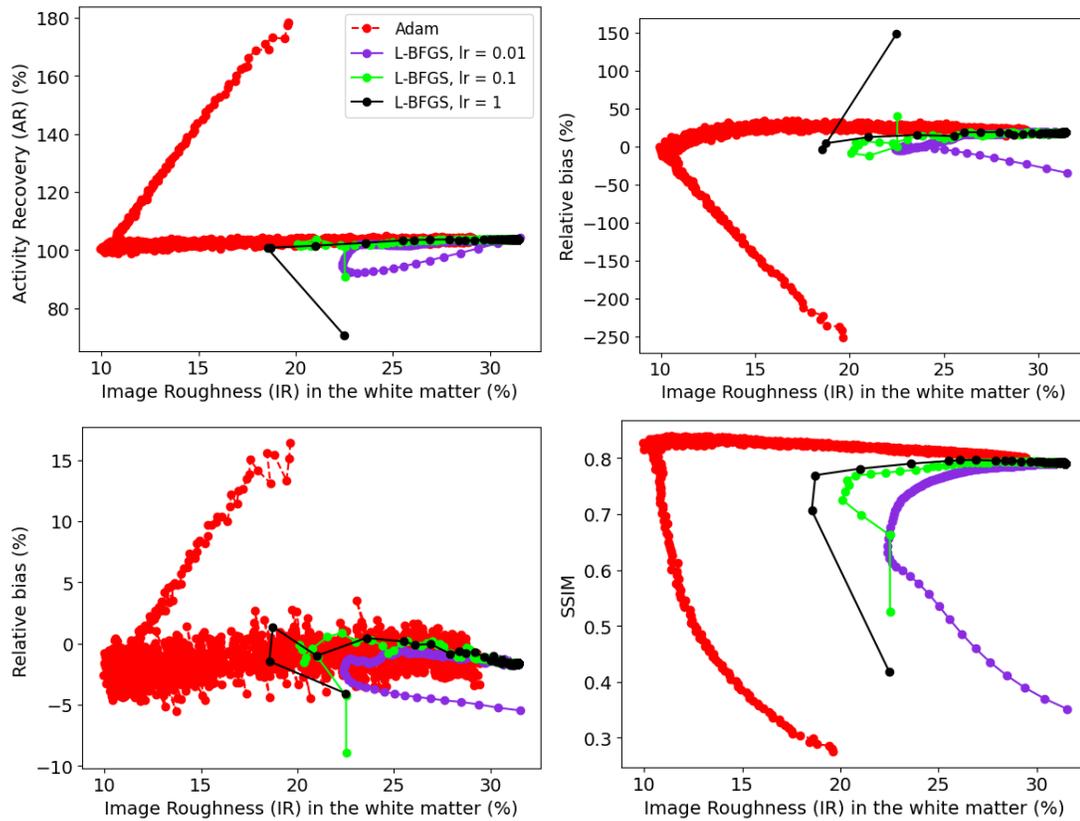


FIGURE 3.15 – Courbes de compromis moyennées sur 20 réplicats sur le fantôme cérébral du chapitre 2 dans la ROI de la tumeur (*haut gauche*), la ROI du LCR (*haut droite*), la ROI de la matière blanche (*bas gauche*) et le SSIM dans l'ensemble du cerveau (*bas droite*). Les courbes montrent l'évolution des métriques au cours des itérations en fonction de l'optimiseur choisi. Les 100 premières itérations de Adam et respectivement les 30,6,1 premières itérations de L-BFGS ont été retirées du graphe pour la lisibilité (respectivement pour valeurs de taux d'apprentissage 0,01,0,1,1).

3.5.1.3 Résultats

La figure 3.15 montre les courbes de compromis obtenues dans le fantôme cérébral du chapitre 2 lors du débruitage DIP avec Adam ou L-BFGS pour différentes valeurs de taux d'apprentissage. L-BFGS avec un taux d'apprentissage élevé permet de faire évoluer la courbe de compromis en seulement quelques itérations. Même si le taux d'apprentissage le plus élevé que nous avons essayé permet d'obtenir la meilleure courbe de compromis avec L-BFGS, celle-ci est moins bonne que celle obtenue avec Adam qui propose d'aussi bonnes métriques pour une rugosité de l'image beaucoup plus faible. De plus, plus le taux d'apprentissage de L-BFGS est élevé, plus L-BFGS converge rapidement vers l'image étiquette.

3.5.1.4 Discussion

L'optimiseur L-BFGS ne permet pas d'obtenir une image de sortie du DIP avec une aussi faible rugosité de l'image que lorsque l'optimiseur Adam est utilisé. De plus, la meilleure courbe de compromis de L-BFGS est obtenue pour le plus grand taux d'apprentissage essayé, mais évolue très rapidement et manque possiblement de meilleurs images que le DIP pourrait produire.

3.5.1.5 Conclusion

L'utilisation d'Adam avec le DIP permet d'obtenir de meilleurs résultats que l'utilisation de L-BFGS. Les images produites avec Adam par le réseau DIP peuvent avoir une plus faible rugosité de l'image qu'avec L-BFGS. L'optimiseur Adam a été utilisé dans le chapitre 2.

3.5.2 Méthodes de mise à l'échelle pour le DIP

La mise à jour des poids est une opération sensible dans les réseaux de neurones profonds. En effet, le gradient peut exploser ou au contraire devenir quasiment nul dûs aux multiplications successives effectuées lors de l'opération de rétropropagation¹. Cela peut empêcher l'optimisation du réseau de neurones ou bien conduire à une divergence des poids. La mise à l'échelle de l'image d'entrée et de l'image étiquette du DIP est alors utile pour stabiliser l'optimisation du réseau [37]. De plus, l'utilisation de méthodes de mise à l'échelle permettra une meilleure adaptation des résultats de ce chapitre à d'autres fantômes.

3.5.2.1 Expériences

Pour une image x , nous utiliserons les notations suivantes :

La moyenne de x sera notée $\mu_x = \sum_{j=1}^J x_j$. Son écart-type sera notée $\sigma_x = \sqrt{\frac{1}{J} \sum_{j=1}^J (x_j - \mu_x)^2}$.

Son minimum et son maximum seront notées respectivement m_x et M_x .

Nous avons étudié trois méthodes de mise à l'échelle de l'image d'entrée et de l'image étiquette :

- Standardisation - l'image x subit la transformation affine suivante, rendant sa distribution de voxels de moyenne 0 et de variance 1 :

$$x_{\text{stand}} \rightarrow \frac{x - \mu_x}{\sigma_x} \quad (3.19)$$

- Normalisation - l'image x subit la transformation affine suivante, dont les valeurs de voxels ont un minimum de 0 et un maximum de 1 :

$$x_{\text{norm}} \rightarrow \frac{x - m_x}{M_x - m_x} \quad (3.20)$$

1. La rétropropagation permet de calculer le gradient de la fonction de coût par rapport à chaque poids du réseau. Elle exploite la règle de la chaîne pour calculer des dérivées de fonctions composées. Le nombre de multiplications à effectuer augmente avec la profondeur du réseau de neurones.

- Normalisation au maximum (que nous avons appelé positive normalization en anglais) - l'image x subit la transformation affine suivante et devient de minimum $\frac{\min_x}{\max_x}$ et de maximum 1 :

$$x_{\text{norm_max}} \rightarrow \frac{x}{M_x} \quad (3.21)$$

A l'issu du débruitage DIP, les images subissent une mise à l'échelle dans l'autre sens, ayant pour but de ramener l'image de sortie du DIP dans le même espace que les images reconstruites pour être utilisé à la prochaine itération externe. Les opérations suivantes sont effectuées :

- Déstandardisation :

$$x_d \rightarrow \sigma_x x_{\text{stand}} + \mu_x \quad (3.22)$$

- Dénormalisation :

$$x_d \rightarrow (M_x - m_x)x_{\text{norm}} + m_x \quad (3.23)$$

- Dénormalisation au maximum :

$$x_d \rightarrow M_x * x_{\text{norm_max}} \quad (3.24)$$

D'une part, pour la standardisation et la normalisation, si l'image x_{stand} ou x_{norm} est produite par une architecture avec ReLU, alors le minimum de l'image de sortie du DIP peut être zéro, mais x_d aura alors un minimum non nul (valant μ_x si standardisation ou \min_x si normalisation). D'autre part, si $x_{\text{norm_max}}$ est produit par une architecture avec ReLU, alors le minimum de $x_{\text{norm_max}}$ peut être zéro, ce qui sera le cas aussi pour x_d .

Nous avons utilisé les simulations FDG du fantôme cérébral du chapitre 2 et avons fait des reconstructions avec les algorithmes DIPRecon et DNA en faisant varier le type de mise à l'échelle (standardisation, normalisation, normalisation au maximum). Chaque méthode a été lancée pour 650 itérations externes.

3.5.2.2 Résultats

La figure 3.16 montre les courbes de compromis entre les métriques étudiées et la rugosité de l'image pour différentes méthodes de mise à l'échelle appliquées au DNA et au DIPRecon². Pour une même rugosité de l'image, l'algorithme DIPRecon obtient un SSIM et une récupération d'activité dans la région chaude légèrement supérieure avec standardisation plutôt que normalisation au maximum. Cependant, dans la région froide, le DIPRecon est biaisé positivement avec la mise à l'échelle standardisation, plus particulièrement pour des valeurs de rugosité de l'image proches de celle atteintes par le DNA. A l'inverse, le DIPRecon avec normalisation au maximum est biaisé négativement (comme déjà observé dans les résultats du chapitre 2, partie 2.5.2). Quant au DNA, la méthode de normalisation au maximum donne des résultats quasiment identiques à celle de la normalisation. La courbe du DNA avec standardisation démarre avec une rugosité de l'image plus élevée. Cette méthode de mise à l'échelle permet tout de même au DNA d'effectuer une réduction de biais dans la région froide, tout en obtenant une récupération d'activité proche de 100%. Seul le SSIM du DNA avec standardisation est moins bon qu'avec les autres méthodes. Enfin, tous les algorithmes ont un biais relatif dans la matière blanche autour de 0%.

3.5.2.3 Discussion

La méthode de standardisation obtient des résultats différents des deux autres méthodes de mises à l'échelle étudiées pour l'algorithme DNA. En effet, l'initialisation du DNA avec standardisation est plus bruitée qu'avec les deux autres méthodes. Dans la partie 2.4.3, nous avons

². La normalisation appliquée au DIPRecon n'est pas montrée car nous n'avons pas réussi à faire fonctionner l'implémentation au moment de l'étude.

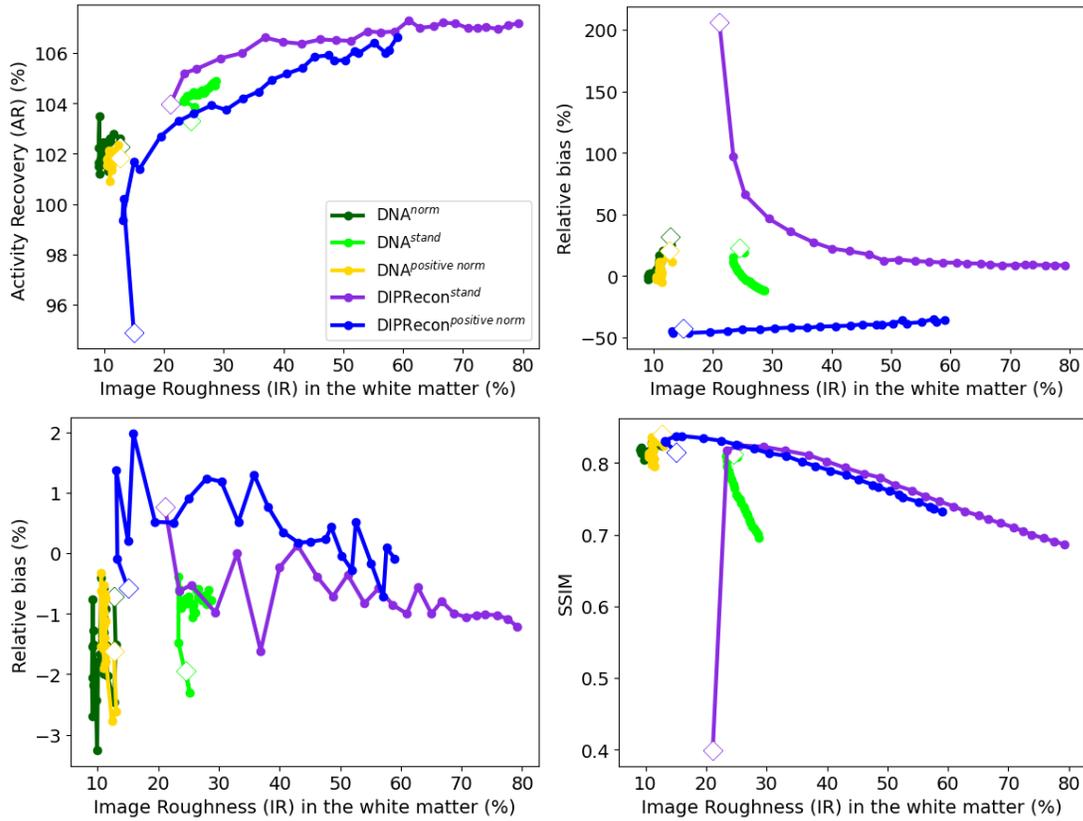


FIGURE 3.16 – Courbes de compromis moyennées sur 15 réplicats sur le fantôme cérébral FDG dans la ROI de la tumeur (*haut gauche*), la ROI du LCR (*haut droite*), la ROI de la matière blanche (*bas gauche*) et le SSIM dans l'ensemble du cerveau (*bas droite*), pour l'algorithme DNA proposé et le DIPRecon, en faisant varier la méthode de mise à l'échelle (standardisation, normalisation ou normalisation au maximum, repérées par les abréviations stand, norm et positive norm sur le graphe). Les diamants blancs correspondent à l'initialisation DIP-BSREM pour chaque algorithme. Certains réplicats du DNA et du DIPRecon ne sont pas montrés car l'algorithme EMV n'a pas détecté de minimum.

remarqué que la standardisation permettait d’atteindre le pic de MSE en moins d’itérations que la normalisation au maximum. Ce pic avait bien été repéré par l’algorithme EMV. Cependant, dans cette partie, l’EMV manque le pic de MSE et propose une itération dont l’image correspondante a été très peu débruitée par le DIP. C’est d’ailleurs encore plus le cas pour le DIPRecon avec standardisation dont l’initialisation a un SSIM très faible comparé aux autres méthodes. Les paramètres de facteur de décroissance et de nombre de patience de l’EMV devraient être réglés différemment pour le DIP avec standardisation par rapport à la normalisation au maximum.

D’une part, la standardisation rend le DIPRecon intrinsèquement biaisé positivement à cause de l’ajout de l’écart-type à une image non-négative produite par l’architecture sans ReLU. D’autre part, la normalisation au maximum rend le DIPRecon biaisé négativement car une région non nulle dans l’image étiquette mais mise à zéro par la ReLU (cf. partie 2.5.1) reste à zéro après dénormalisation au maximum.

3.5.2.4 Conclusion

La normalisation au maximum et la normalisation appliquées au DNA permettent un meilleur débruitage DIP que la standardisation avec les paramètres choisis de l’algorithme EMV. Quant au DIPRecon, la standardisation apporte du biais positif contrairement à la normalisation au maximum qui apporte du biais négatif dans les régions froides. La normalisation au maximum a été utilisée dans le chapitre 2 pour le DNA et le DIPRecon.

3.5.3 Influence du paramètre ρ_1

Dans cette partie, nous cherchons à étudier l’influence du paramètre de pénalité ADMM ρ_1 . Ce paramètre est commun aux algorithmes DIPRecon et DNA. La politique adaptative proposée par [104] utilisée pour mettre à jour le paramètre de pénalité ADMM ρ_2 n’est pas applicable directement pour mettre à jour ρ_1 . En effet, les résidus dans [104] sont définis pour un problème d’optimisation avec contrainte linéaire, ce qui n’est pas le cas ici avec le réseau DIP. Une dérivation mathématique doit être menée pour déterminer les résidus associés à la contrainte du DIP dans les algorithmes DNA et DIPRecon. Cependant, la méthode des résidus relatifs cherche à faire converger les deux résidus impliquant une convergence de l’algorithme. Mais le DNA et le DIPRecon ne doivent pas être amenés à convergence car leur comportement serait alors respectivement comme ADMM-Reg et MLEM sans régularisation (cf. partie 2.6). Dans cette thèse, nous avons plutôt étudié de façon heuristique l’influence du paramètre de pénalité ρ_1 sur la reconstruction sans chercher encore à l’automatiser.

3.5.3.1 Expériences

Nous avons utilisé les simulations FDG du fantôme cérébral du chapitre 2, et fait varier les valeurs de ρ_1 dans $[3 \times 10^{-5}; 3]$ pour la reconstruction. Nous avons utilisé l’algorithme DNA-APPGML pour mieux observer la différence de rugosité de l’image en fonction des valeurs de ρ_1 par rapport à l’algorithme DNA (cf. partie 3.4.2.2). Le nombre d’itérations internes a été varié dans un second temps (1, 10 et 30 itérations internes) pour étudier conjointement son influence avec l’hyperparamètre ρ_1 . 300 itérations externes ont été réalisées dans chaque cas. Une borne $A = -10$ a été utilisée comme dans la partie 3.4.2.3.1.

3.5.3.2 Résultats

La figure 3.17 montre le comportement quantitatif du DNA-APPGML en fonction des valeurs de ρ_1 , pour un nombre fixe d’itérations externes. Tout d’abord, la rugosité de l’image augmente plus vite en diminuant la valeur de ρ_1 . Les différences se voient principalement entre $\rho_1 = 3$, $\rho = 0.3$ et $\rho = 0.03$. Il y a très peu de différence en continuant de diminuer la valeur de ρ_1 en dessous de 0.03 pour ce fantôme. De plus, pour toutes les métriques étudiées, les différentes courbes empruntent le même chemin, si on ignore la variabilité entre chaque point d’une même courbe. Cependant, nous observons que pour $\rho_1 = 3$, l’algorithme a une tendance légèrement

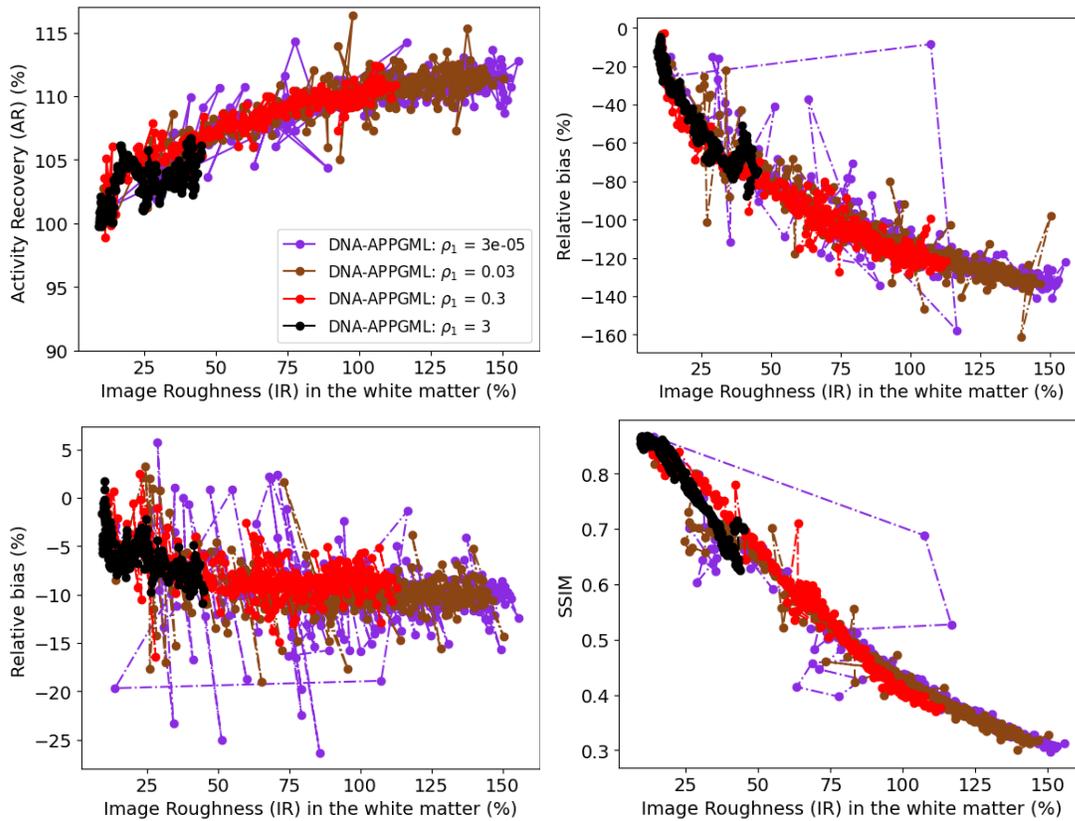


FIGURE 3.17 – Courbes de compromis pour 1 réplicat pour le fantôme cérébral FDG, dans la ROI de la tumeur (*haut gauche*), la ROI du LCR (*haut droite*), la ROI de la matière blanche (*bas gauche*) et le SSIM dans l'ensemble du cerveau (*bas droite*). L'algorithme DNA-APPGML a été utilisé avec 10 itération internes et différentes valeurs de ρ_1 .

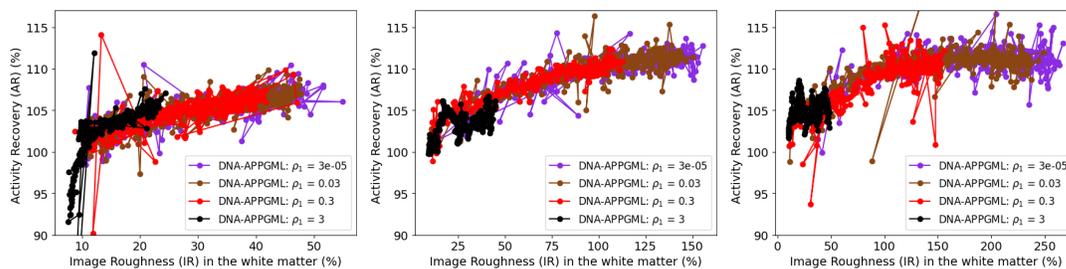


FIGURE 3.18 – Courbes de compromis pour 1 réplicat pour le fantôme cérébral FDG, dans la ROI de la tumeur. L'algorithme DNA-APPGML a été utilisé avec 1, 10 ou 30 itérations internes (respectivement à gauche, au milieu et à droite). Les axes des abscisses ne sont pas à la même échelle.

différente, avec des métriques légèrement moins monotone (en ignorant la variabilité entre chaque point d'une même courbe).

La figure 3.18 montre l'effet du nombre d'itérations internes sur le compromis entre récupération d'activité dans la tumeur et rugosité, pour différentes valeurs de ρ_1 . Nous observons toujours une augmentation plus rapide de la rugosité de l'image pour un paramètre de pénalité ρ_1 faible, pour les trois nombres d'itérations internes testés. A ρ_1 fixé, la rugosité de l'image augmente plus vite lorsque le nombre d'itérations internes est plus élevé. Ce phénomène est plus important pour des valeurs de ρ_1 faibles.

3.5.3.3 Discussion

[59] souligne que la valeur du paramètre de pénalité ADMM n'influe pas sur la solution obtenue à convergence. Nous remarquons avec cette étude que ce paramètre n'influe pas non plus sur le chemin emprunté par l'algorithme. En revanche, pour des valeurs trop fortes de ρ_1 , le comportement peut être un peu plus chaotique, jusqu'à diverger (nombres infinis dans l'image pour $\rho_1 \geq 30$). Tout le comportement décrit précédemment est attendu théoriquement, étant donné que le paramètre $\frac{\rho_1}{2}$ peut être interprété dans l'équation (2.15a) du chapitre 2 comme un paramètre de pénalité d'une reconstruction PML. Le terme de pénalité $\frac{\rho_1}{2} \|x - f(\theta^n|z) + \mu^n\|_2^2$ contrôle la confiance que l'on a dans le débruitage réalisé par le DIP à l'itération externe précédente. Un paramètre ρ_1 faible va donc favoriser la reconstruction d'une image, en prenant faiblement en compte l'image de sortie du DIP. L'image reconstruite dans l'équation (2.15a) sera donc plus bruitée que pour ρ_1 fort.

De la même façon, l'augmentation du nombre d'itérations internes entraîne une augmentation plus rapide de la rugosité de l'image. Cela est cohérent avec la théorie qui prévoit que le bruit augmente de plus en plus au cours des itérations pour un algorithme ML, étant donné la nature mal posée du problème inverse TEP. De plus, ce phénomène est plus important pour des valeurs de ρ_1 faibles, car la convergence est atteinte avec moins d'itérations. En effet, l'algorithme ADMM-Reg nécessite plus d'itérations pour des paramètres de pénalité faibles pour atteindre la convergence, pour une même valeur du critère d'arrêt donné dans [12]. Ajouter des itérations internes a donc moins d'impact pour des valeurs de ρ_1 fortes.

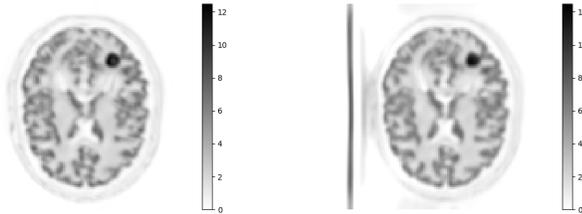


FIGURE 3.19 – Images du fantôme cérébral du chapitre 2 reconstruites avec le DNA, pour $\rho_1 = 3$ et 10 itérations internes. L'image de *gauche* est reconstruite après 3 itérations externes et celle de *droite* après 5 itérations externes.

D'après les résultats obtenus, il est essentiel de bien choisir le paramètre ρ_1 ainsi que le nombre d'itérations internes lors de l'utilisation du DNA. En effet, avec un ρ_1 faible et un nombre d'itérations internes élevés, une augmentation rapide du bruit est favorisée. Dans ce cas, avec le DNA (ADMM-Reg), nous avons observé l'apparition d'artefacts comme illustré dans la figure 3.19. Pour expliquer ce phénomène, nous avons observé que l'algorithme ADMM-Reg reconstruit des images avec une structure de bruit qui lui est propre (cf. figure 3.20). Nous supposons que si ces structures sont trop importantes, alors le DIP peut les reconnaître comme une vraie caractéristique de l'image à débruiter, ce qui crée ces lignes. Ainsi, un nombre d'itérations d'ADMM-Reg limité en combinaison avec un paramètre de pénalité ρ_1 favorisant le DIP plutôt que le terme d'attache aux données peut empêcher le DIP de reconnaître des structures dans le bruit induit par l'ADMM-Reg. Pour compléter, ces artefacts ont été observés

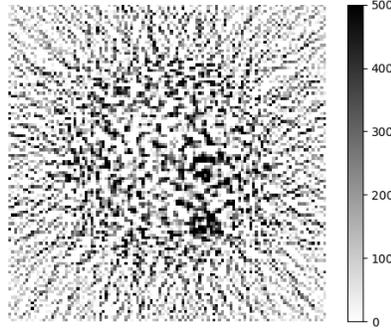


FIGURE 3.20 – Image du fantôme cylindrique 2 reconstruite après 1000 itérations avec l’algorithme ADMM-Reg sans régularisation.

moins souvent et de façon moins prononcée dans les simulations ^{90}Y . La différence majeure avec le cas FDG est la proportion de coïncidences fortuites venant des cristaux LSO/LYSO qui est plus importante. Nous supposons que la structure du bruit en est modifiée et est moins reconnaissable par le DIP dans le cas ^{90}Y .

3.5.3.4 Conclusion

Les images produites par le DNA-APPGML sont bruitées plus rapidement avec un paramètre ρ_1 faible. De plus, le choix de ρ_1 doit être effectué conjointement avec le nombre d’itérations internes. En particulier dans nos simulations type FDG, le DNA a été utilisé avec $\rho_1 < 3$ et 3 itérations internes. D’autres valeurs ont conduit à des artefacts. Ce paramétrage a été utilisé pour le DNA dans le chapitre 2.

3.5.4 Influence du taux d’apprentissage

[92] a proposé l’idée du DIP en observant dans leurs expériences que le DIP avait plus de mal à reproduire une image bruitée par rapport à une image naturelle. De plus, les auteurs dans [19; 83] ont montré que le DIP possède un biais spectral, c’est-à-dire qu’il est enclin à reproduire d’abord les fréquences les plus faibles jusqu’aux fréquences les plus élevées au fur et à mesure des itérations. Les observations de [92] sont donc justifiées par ce biais spectral, le bruit utilisé dans leurs expériences étant de hautes fréquences.

Dans cette partie, nous nous intéressons uniquement à l’optimiseur Adam. Nous introduisons tout d’abord le concept de correspondance en bande de fréquence qui nous a servi pour étudier l’influence du taux d’apprentissage.

3.5.4.1 Correspondance en bande de fréquence (FBC)

La correspondance en bande de fréquence (FBC - Frequency Band Correspondence) a été proposée dans [83] dans le cadre du DIP. Elle cherche à mesurer quelles fréquences d’une image étiquette sont récupérées par la sortie d’un réseau DIP.

Pour cela, considérons une image x_{label} à débruiter par le DIP. Soit θ^t les poids du réseau DIP et $f(\theta^t|z)$ l’image de sortie du réseau DIP, tout deux à l’itération t . On calcule les transformées de Fourier \mathcal{F} de x_{label} et $f(\theta^t|z)$ pour obtenir une représentation en fréquences de ces deux images. En pratique, la transformée de Fourier discrète est utilisée.

Les auteurs dans [83] définissent la quantité \mathcal{H}^t à l’itération t pour le pixel j comme le ratio élément par élément des modules des deux transformées de Fourier :

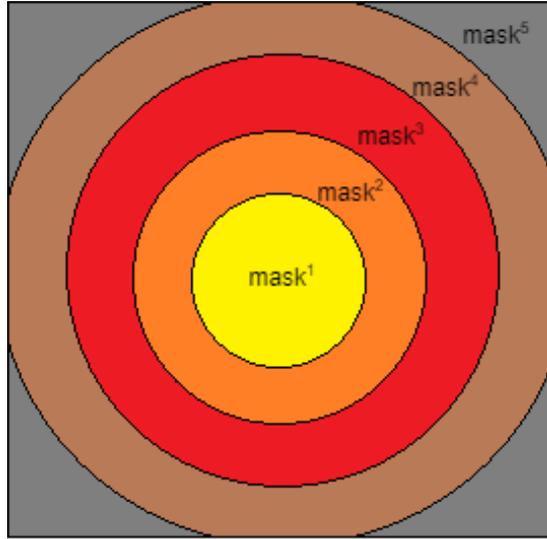


FIGURE 3.21 – Masques appliqués sur la transformée de Fourier d'une image. Ces masques sont utilisés dans le calcul de la FBC.

$$\mathcal{H}_j^t = \frac{|\mathcal{F}\{f(\theta^t|z)\}|_j}{|\mathcal{F}\{x_{\text{label}}\}|_j}. \quad (3.25)$$

Ce ratio est saturé à 1 pour des valeurs de $|\mathcal{F}\{f(\theta^t|z)\}|_j > |\mathcal{F}\{x_{\text{label}}\}|_j$. La FBC est en pratique calculée pour P bandes de fréquences comme présenté dans la figure 3.21. Des masques binaires $\text{mask}^{(p)}$ sont définis pour chaque bande de fréquences. [83] calcule finalement la FBC \mathcal{H}_p^t pour la bande de fréquences p et à l'itération t comme la somme de \mathcal{H}^t à l'intérieur du masque p divisé par le nombre d'éléments du masque :

$$\mathcal{H}_p^t = \frac{\sum_{j \in \text{mask}^{(p)}} \mathcal{H}_j^t}{\text{Card}\{\text{mask}^{(p)}\}}. \quad (3.26)$$

La FBC ainsi définie est toujours entre 0 et 1 pour chaque bande de fréquence p .

Dans notre étude, plutôt que de saturer des valeurs de transformée de Fourier, nous proposons de calculer la FBC pour la bande de fréquences p à l'itération t comme suit :

$$\text{FBC}_p^t = \frac{\sum_{j \in \text{mask}^{(p)}} |\mathcal{F}\{f(\theta^t|z)\}|_j |\mathcal{F}\{x_{\text{label}}\}|_j}{\sqrt{\sum_{j \in \text{mask}^{(p)}} |\mathcal{F}\{f(\theta^t|z)\}|_j^2 \sum_{j \in \text{mask}^{(p)}} |\mathcal{F}\{x_{\text{label}}\}|_j^2}}. \quad (3.27)$$

Par l'inégalité de Cauchy-Schwarz, la FBC définie ainsi est naturellement à valeur dans $[0;1]$ et représente une mesure de similarité entre $|\mathcal{F}\{f(\theta^t|z)\}|$ et $|\mathcal{F}\{x_{\text{label}}\}|$.

3.5.4.2 Expériences

Nous avons utilisé les simulations FDG du fantôme cylindrique 3, et reconstruits des images légèrement bruitées avec l'algorithme BSREM+RDP. Plusieurs débruitages avec un réseau DIP ont été réalisés avec l'optimiseur Adam, en faisant varier les valeurs de taux d'apprentissage γ dans l'intervalle $[10^{-5}; 5]$. Nous avons représenté la FBC $_p^t$ en fonction des itérations t du DIP. Nous séparons les fréquences reproduites par le DIP en $P = 5$ bandes de fréquences comme dans [83], des plus basses aux plus hautes : lowest, low, medium, high, highest.

3.5.4.3 Résultats

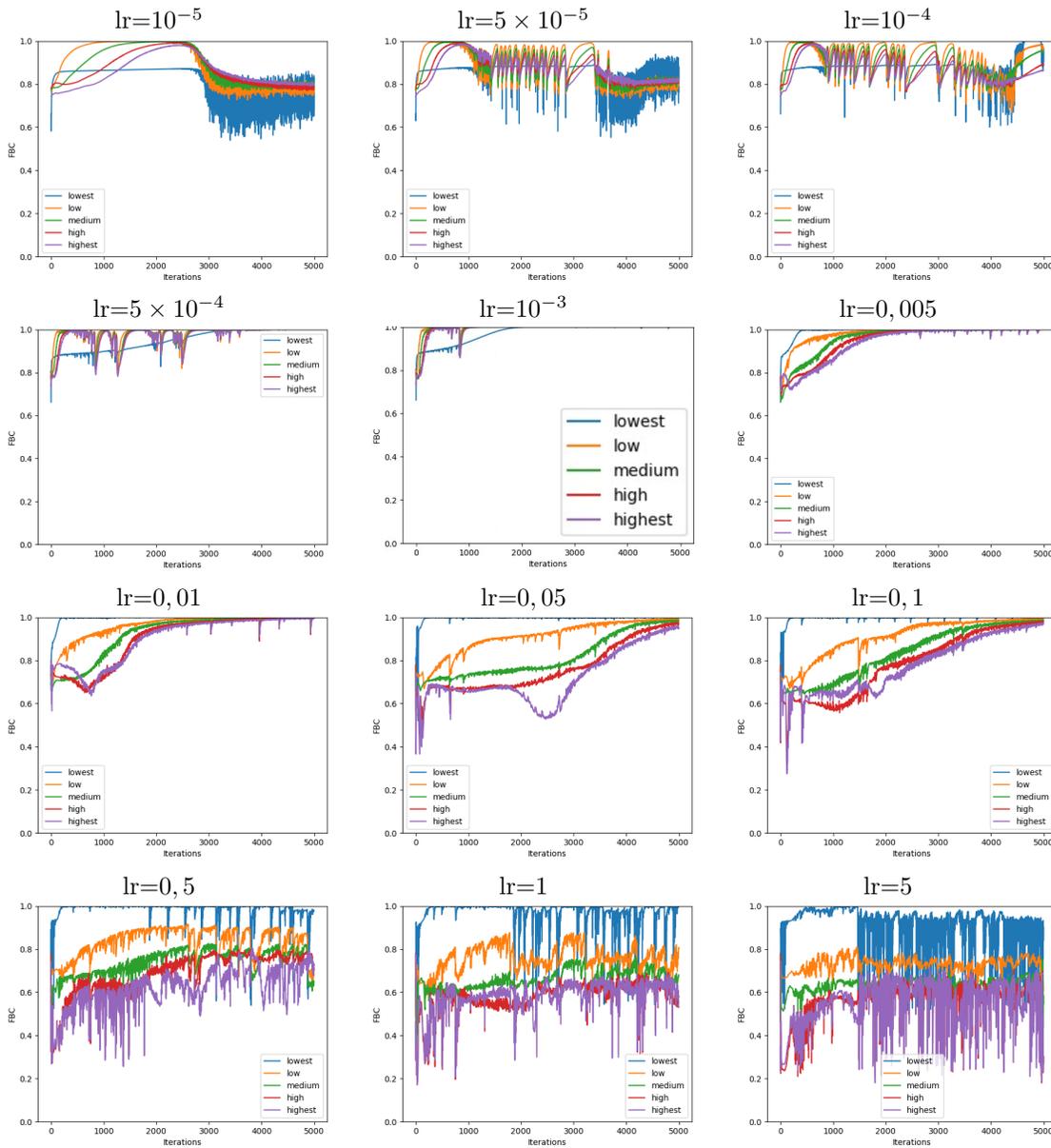


FIGURE 3.22 – FBC des images en sortie du DIP en fonction des itérations. L'axe des ordonnées représente la FBC entre 0 et 1. L'axe des abscisses représente les itérations de 0 à 5000. Chaque sous figure correspond aux résultats obtenus pour un taux d'apprentissage γ différent.

La figure 3.22 présente les FBCs obtenues pour 12 valeurs différentes de taux d'apprentissage γ jusqu'à 5000 itérations du DIP, et la figure 3.23 montre quelques images obtenues pour certaines valeurs de taux d'apprentissage après 500 itérations.

Dans la figure 3.22, nous observons que les trois plus petites valeurs de taux d'apprentissage ($\gamma \cdot 10^{-4}$) ne permettent pas au réseau DIP de reproduire les plus basses fréquences de l'image étiquette x_{label} , même au bout des 5000 itérations.

Les deux valeurs suivantes ($\gamma = 5 \cdot 10^{-4}$ et $\gamma = 10^{-3}$) permettent au réseau DIP de reproduire l'image étiquette en moins de 5000 itérations, les FBCs pour les 5 valeurs de fréquence atteignant toutes 1. Cependant, les fréquences les plus basses (lowest) sont les plus lentes à être intégrées à l'image de sortie du DIP. En regardant les images dans le régime où ces fréquences ne sont

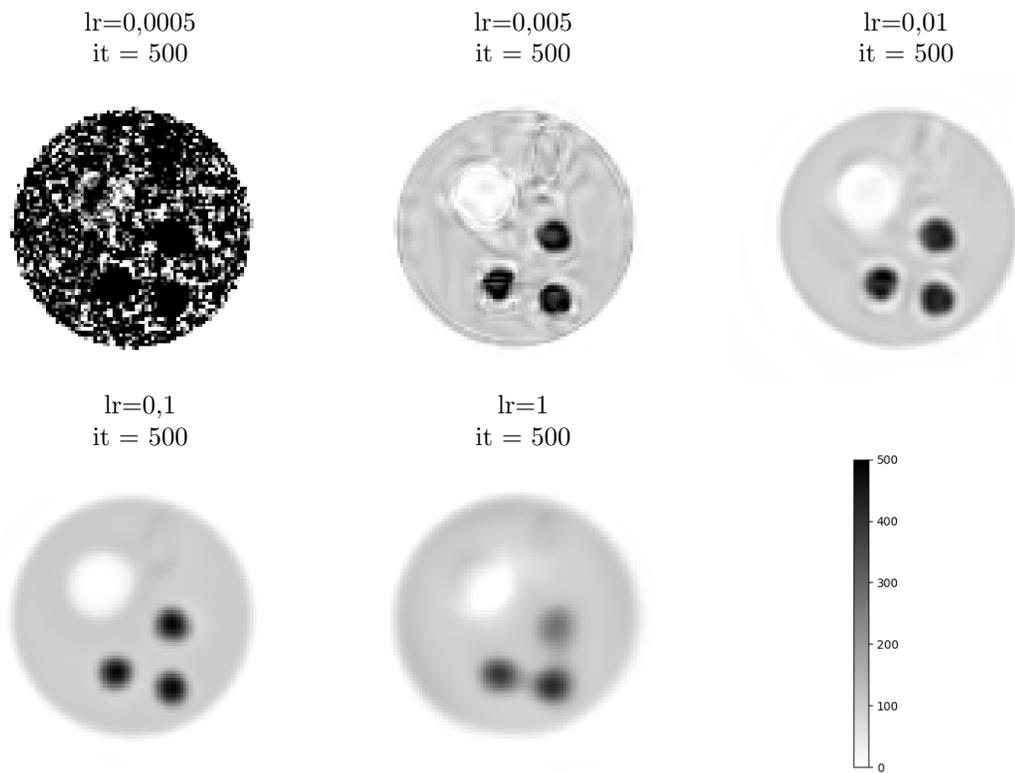


FIGURE 3.23 – Images produites par le réseau DIP pour différentes valeurs de taux d'apprentissage.

pas encore complètement dans l'image de sortie (après 500 itérations par exemple pour $\gamma = 5 \cdot 10^{-4}$ comme présenté dans la figure 3.23), celles-ci sont visuellement aussi bruitées que l'image étiquette. En revanche, les moyennes locales dans différentes ROIs du fantôme sont autour de 10 fois supérieures aux moyennes de la GT. La moyenne de l'image de sortie du DIP, qui est la fréquence nulle de l'image n'est pas bonne tant que les courbes bleues (correspondant aux plus basses fréquences) pour ces valeurs de taux d'apprentissage n'ont pas encore atteint la valeur de 1.

Pour des valeurs de taux d'apprentissage au-dessus de 0,005, les plus basses fréquences sont les premières à être incorporées dans la sortie du DIP. Pour un taux d'apprentissage de 0,005, la courbe des plus basses fréquences (bleue) atteint 1 autour de 500 itérations. Mais à ce nombre d'itérations, on voit dans la figure 3.23 que l'image correspondante contient déjà un peu trop de hautes fréquences non présentes dans la GT.

Les valeurs de taux d'apprentissage de 0,01 et 0,05, 0,1 permettent d'incorporer progressivement les fréquences des plus basses aux plus hautes. De plus, les plus basses fréquences sont introduites en très peu d'itérations, plus rapidement que pour les valeurs plus faibles de taux d'apprentissage. Après quelques centaines d'itérations pour un taux d'apprentissage de 0,01, l'image de sortie du DIP est assez proche visuellement de la GT (cf. figure 3.23). Pour les valeurs de 0,05 et 0,1, les fréquences moyennes (low et medium dans la figure 3.22) prennent plus de temps à apparaître dans la sortie du DIP. Pour 500 itérations avec un taux d'apprentissage de 0,1, l'image de sortie montrée dans la figure 3.23 est encore assez lisse avec peu de hautes fréquences.

Enfin, nous remarquons que les trois plus grandes valeurs de taux d'apprentissage conduisent à un comportement assez chaotique du DIP, qui n'arrive jamais à reproduire parfaitement l'image étiquette comme demandé par la fonction de coût. Seules les fréquences les plus basses apparaissent dans la sortie du DIP (cf. figure 3.23 pour $\gamma = 0,1$).

3.5.4.4 Discussion

Le DIP a donc un comportement différent en fonction des valeurs de taux d'apprentissage. Pour des valeurs trop faibles, celui-ci va être lent à converger, et va reproduire les fréquences les plus basses de l'image étiquette après toutes les autres. Ce comportement n'est pas celui décrit par [92], et n'est pas adapté pour débruiter une image. Les valeurs de taux d'apprentissage trop élevées ne sont pas non plus utilisables, d'une part étant donné le comportement chaotique du DIP à chaque itération, d'autre part du fait que le DIP n'arrive pas à récupérer toutes les fréquences de l'image étiquette. Dans notre cas, la valeur optimale de taux d'apprentissage est de 0,01, qui permet de récupérer les fréquences de l'image étiquette des plus faibles aux plus élevées, avec une image optimale obtenue après quelques centaines d'itérations du DIP.

3.5.4.5 Conclusion

Le choix du taux d'apprentissage est essentiel pour permettre au DIP d'avoir un biais spectral. Dans nos expériences, un taux d'apprentissage de 0,01 a permis d'obtenir le biais spectral attendu ainsi qu'une image débruitée de qualité. Cette valeur de taux d'apprentissage a été utilisée dans le chapitre 2.

3.5.5 Étude statistique sur les sources de variabilité intervenant dans un débruitage DIP

Cette partie ne concerne pas l'étude d'un hyperparamètre de la figure 3.1. Ici, nous avons voulu étudier la robustesse statistique du DIP face à différentes sources de bruit inhérentes à son optimisation. En effet, les poids du réseau de neurones ont été jusqu'alors choisis aléatoirement. L'entrée peut elle aussi être choisie aléatoirement comme proposé initialement par [92] (cf. partie 2.2.2). Enfin, une image reconstruite correspond à un réplicat bruité des données TEP Poisson.

3.5.5.1 Expériences

Dans toute cette étude, pour étudier quantitativement les performances du DIP, nous avons calculé à chaque itération du DIP la MSE entre les images en sortie du réseau DIP et la GT. Pour chaque débruitage DIP, nous nous intéressons à la valeur minimale de MSE atteinte au cours des itérations, notée MSE peak. Des boîtes à moustaches ont été réalisées pour représenter la distribution du MSE peak pour 100 débruitages DIP effectués.

Nous avons effectué des débruitages DIP d’une image étiquette reconstruite par BSREM+RDP. Le fantôme cylindrique 3 a été utilisé avec des simulations ^{90}Y . Les mêmes hyperparamètres que ceux choisis dans la partie 2.7.2.2 pour reconstruire les images BSREM de départ ont été utilisés. Les débruitages DIP ont été effectués avec le même paramétrage que dans la partie 2.4.2.2, excepté la standardisation qui a été utilisée comme méthode de mise à l’échelle. Par défaut, l’initialisation est réalisée avec la méthode de Kaiming [49] avec distribution uniforme.

3.5.5.1.1 Expérience 1 Nous avons tout d’abord étudié la variabilité statistique associée au réplicat de Poisson. En effet, la variabilité statistique des données Poisson entraîne une variabilité statistique sur les images TEP reconstruites.

Nous avons considéré 100 images reconstruites par BSREM+RDP pour 100 réplicats différents. 100 optimisations du DIP ont été réalisées pour débruiter ces 100 images.

De même que pour les images de sortie du DIP, nous avons calculé la MSE entre l’image BSREM à débruiter et la GT, pour les 100 réplicats. Cette distribution de MSE est représentée par une boîte à moustaches servant de référence et qui sera appelée “baseline” par la suite.

La boîte à moustaches issue des débruitages DIP a été comparée à la baseline pour évaluer la performance du DIP et sa robustesse statistique.

3.5.5.1.2 Expérience 2 Nous avons ensuite étudié des débruitages DIP pour un même réplicat de Poisson. Nous nous sommes concentrés sur deux autres sources de variabilité pouvant affecter l’image débruitée par le réseau DIP :

- l’image d’entrée du réseau DIP peut être choisie aléatoire ou anatomique (cf. partie 2.2.2). Ici, nous avons considéré 100 réalisations de bruit uniforme en entrée du DIP,
- l’initialisation aléatoire des poids du réseau. Dans cette partie, nous avons considéré 100 réalisations de bruit uniforme ou gaussien, associées aux initialisations de Kaiming [49] ou Xavier [36].

Ces deux sources de variabilité ont d’abord été étudiées seules, en fixant l’autre. Puis nous les avons fait varier ensemble au sein d’une même étude à travers 100 débruitages.

Pour évaluer les performances du DIP, des boîtes à moustaches ont été utilisées pour représenter la distribution des MSE peak. Une boîte à moustaches a été utilisé pour chaque source, ainsi que lorsque les deux sources ont été variées simultanément.

3.5.5.2 Résultats

La figure 3.24 montre les boîtes à moustaches pour l’expérience 1 sur les réplicats. La boîte à moustaches “baseline” associée aux 100 images BSREM à débruiter est disjointe de celle associée aux 100 débruitages DIP.

La figure 3.25 montre les boîtes à moustaches pour l’expérience 2. La deuxième ligne de cette figure montre que l’initialisation par défaut est aussi performante que les autres initialisations. De plus, nous observons que les boîtes à moustaches associées aux 100 images d’entrée et aux 100 réalisations aléatoires des poids du réseau sont proches. Ces deux distributions sont légèrement moins étendues que celle associée aux 100 images de réplicats différents (cf. figure 3.24). Enfin, nous notons que la variation simultanée de l’image d’entrée et des poids du réseau n’augmente pas la variabilité des images de sortie.

3.5.5.3 Discussion

La première expérience montre que le DIP parvient toujours à proposer une image au cours de son optimisation avec un meilleur MSE peak que l’image qu’il cherche à débruiter. De plus,

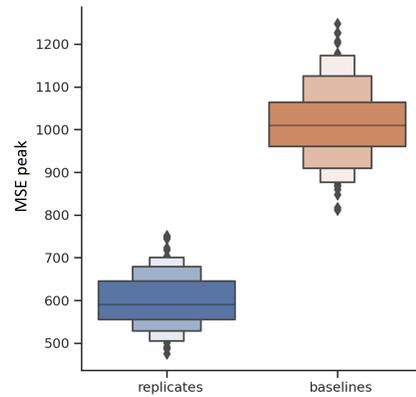


FIGURE 3.24 – Boîtes à moustaches représentant la distribution du MSE peak pour l'expérience 1 : 100 réplicats reconstruits par BSREM+RDP et débruités par le DIP.

les distributions associées à l'image étiquette et à leur débruitage DIP ne se chevauchent pas (cf. figure 3.24).

Ces premières expériences montrent que le côté aléatoire de l'image d'entrée du réseau DIP, des poids du réseau DIP, ou encore du réplicat TEP n'ont pas une influence majeure sur la meilleure image de sortie que propose le réseau DIP, en termes de MSE. Toutes les distributions obtenues se situent en dessous de la MSE de la baseline. De plus, nous concluons que la méthode d'initialisation des poids du réseau n'a pas d'influence sur la meilleure MSE obtenue par la sortie du réseau DIP.

3.5.5.4 Conclusion

La réalisation aléatoire de l'image d'entrée, l'initialisation aléatoire des poids du réseau, ainsi que le réplicat des données TEP sont sources de variabilité au niveau de l'image de sortie du réseau DIP, mais permettent toujours d'obtenir un minimum de MSE meilleur que celle de l'image étiquette.

3.5.6 Étude sur l'architecture du DIP

Cette étude a été réalisée par un stagiaire de l'Ecole Centrale de Nantes dont j'ai participé à l'encadrement, au début de ma troisième année de thèse. Les résultats obtenus pendant le stage n'ont donc pas pu être pris en compte dans l'implémentation du DNA présentée dans le chapitre 2.

De plus, dans cette partie, nous traiterons uniquement de l'influence de l'architecture du réseau DIP pour une tâche de débruitage. L'influence des sauts de connexion et de l'image d'entrée sur la reconstruction avec le DNA sera étudiée spécifiquement dans le chapitre 4.

3.5.6.1 Contexte

Dans le papier originel du DIP [92], les auteurs utilisent un réseau de neurones avec une architecture de type U-Net. Pendant cette thèse, nous avons utilisé un réseau DIP dont l'architecture est inspirée par le U-Net, mais avec quelques modifications proposées par [37] et relatées dans la partie 2.3.3. Cependant, le DIP souffre d'un surapprentissage au bruit s'il est optimisé jusqu'à convergence. Une architecture sous-paramétrée a été proposée dans [50], en opposition à l'architecture originale qui est sur-paramétrée, signifiant qu'elle peut représenter n'importe quelle image étiquette de façon exacte. L'architecture proposée dans [50] tend à réduire le surapprentissage du bruit, mais n'améliore pas la qualité des images produites (en termes de PSNR). Dans [51], les auteurs expliquent que les opérations d'interpolation avec noyaux fixes, comme les opérations d'interpolation bilinéaire sont primordiales. Cela permet à une architecture DIP de reproduire une image naturelle plus rapidement que le bruit. Parallèlement, en traitement

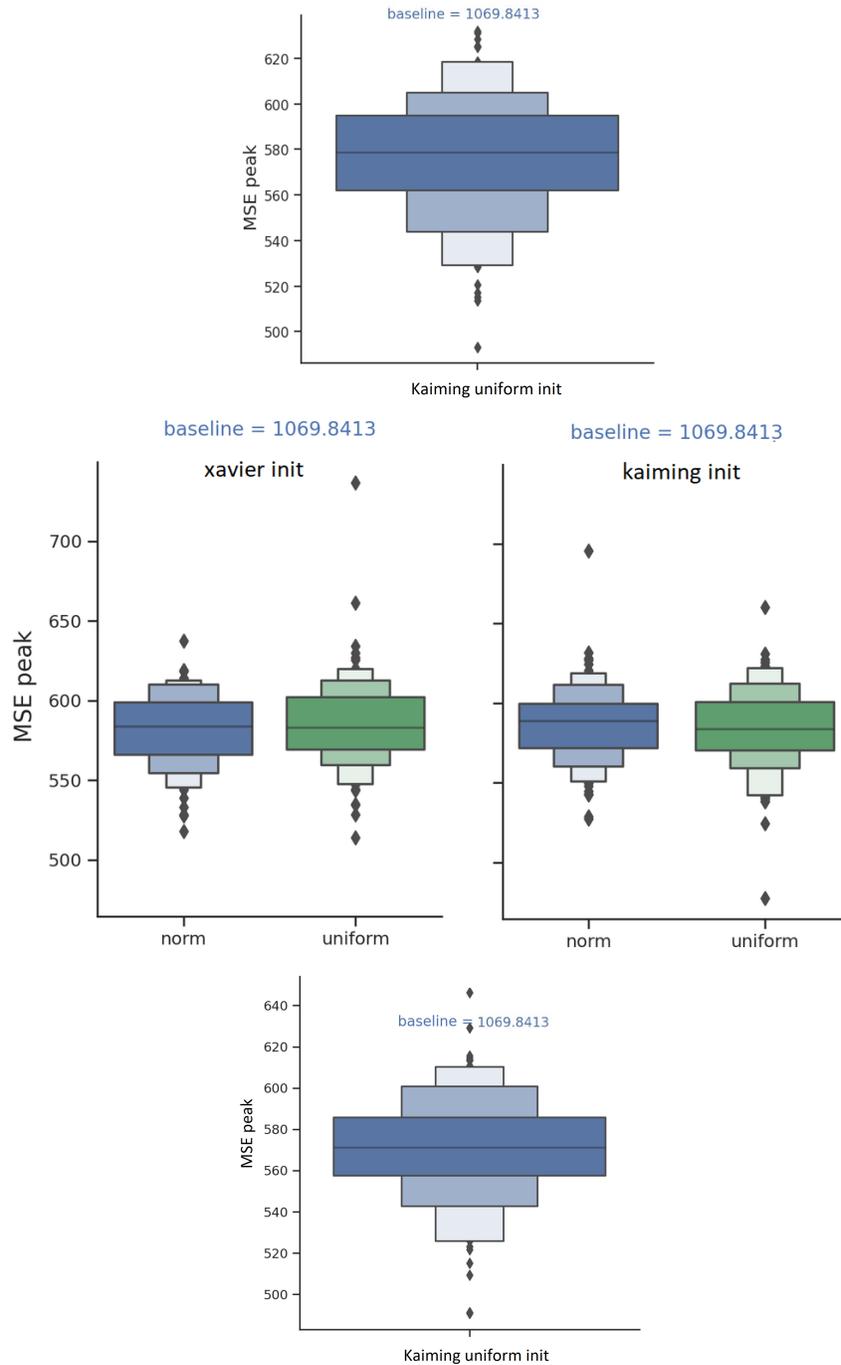


FIGURE 3.25 – Boîtes à moustaches représentant la distribution du MSE peak pour l'expérience 2 : 100 entrées différentes du réseau DIP (*première ligne*), 100 initialisations des poids du réseau DIP pour différentes méthodes (*deuxième ligne*), 100 débruitages avec entrées et poids qui varient simultanément (*troisième ligne*).

d’images, un nouveau type de réseau de neurones (les transformateurs [94]) s’est avéré plus performant que les réseaux de neurones convolutionnels pour l’apprentissage supervisé, mais n’a pas encore été combiné avec le concept de DIP à notre connaissance.

3.5.6.2 Méthodologie générale

Pour chaque partie de cette étude sur l’architecture du DIP, 100 débruitages ont été réalisés afin d’obtenir des distributions statistiques. Pour chaque débruitage, le réplicat TEP de l’image à débruiter est choisi aléatoirement, l’image d’entrée du DIP est choisie aléatoirement et les poids du réseau sont initialisés aléatoirement. Comme dans la partie 3.5.5, le meilleur MSE obtenu au cours d’un débruitage est considéré (MSE peak). Des boîtes à moustaches sont utilisées pour représenter la distribution du MSE peak. L’architecture utilisée dans toute cette étude est l’architecture avec ReLU finale.

3.5.6.3 Étude sur l’architecture d’un réseau convolutionnel pour le DIP

L’idée de cette étude est de voir quelle est la meilleure architecture pour débruiter une image de façon non supervisée. Trois architectures ont notamment été comparées dans [81] : auto encodeur originel utilisé dans [92], décodeur (identique à l’architecture précédente mais sans l’encodeur) et une architecture convolutionnelle sans opération de sur-échantillonnage ou sous-échantillonnage nommée ConvNet dans [81]. Grâce à une analyse avec la FBC (cf. partie 3.5.4.1), les auteurs ont conclu que l’architecture sans opération de sur-échantillonnage récupère trop rapidement les hautes fréquences, bien que d’abord les basses fréquences. Les deux architectures utilisant un décodeur ont un biais spectral comparable. Cependant, le décodeur simple est moins coûteux à entraîner. Les auteurs dans [50] ont aussi insisté sur l’importance des couches de sur-échantillonnage pour assurer une image lisse par morceaux et ainsi empêcher le bruit dans l’image de sortie. De plus, ils ont proposé une architecture appelée Deep Decoder (DD), s’appuyant uniquement sur un décodeur avec des couches convolutionnelles utilisant des noyaux de taille 1×1 (contrairement au U-Net utilisé par [92] qui utilisent des noyaux de taille 3×3).

3.5.6.3.1 Expériences Inspirés par la bibliographie, nous avons comparé l’architecture utilisée pendant cette thèse avec ReLU finale (notée “our w/ SC” dans cette partie), avec une architecture utilisant un décodeur uniquement (notée “our Decoder”) et le DD. Nous avons fait varier la profondeur et le nombre de canaux. Pour une configuration du DD, le nombre de canaux est toujours le même comme dans [50]. En revanche, pour les architectures “our w/ SC” et “our Decoder”, le nombre de canaux démarre à 16 et est doublé après chaque opération de sous-échantillonnage jusqu’à atteindre le nombre de canaux désiré. Ce nombre de canaux est ensuite divisé par 2 après chaque opération de sur-échantillonnage. Par exemple, la figure 2.3 du chapitre 2 est un réseau de type “our w/ SC” avec une profondeur de 3 et 128 canaux.

Dans cette partie et les suivantes, un rembourrage de zéro a été effectué sur l’image étiquette pour obtenir une image de taille 128×128 . Ce nombre de pixels était nécessaire pour pouvoir augmenter le nombre de couches dans les architectures utilisant uniquement un décodeur. La mise à l’échelle utilisée dans ces expériences et les suivantes est la normalisation.

Nous avons utilisés le fantôme cylindrique 3 dans le cadre de simulations ^{90}Y . Les données sont reconstruites avec l’algorithme BSREM+RDP avec les mêmes paramètres que dans la partie 2.7.2.2.

Les images correspondant aux meilleures et aux moins bonnes configurations (en termes de MSE peak) pour chaque architecture ont aussi été comparées.

3.5.6.3.2 Résultats La figure 3.26 présente les boîtes à moustaches obtenues pour les trois architectures d’intérêt, en faisant varier la profondeur et le nombre de canaux. Nous remarquons tout d’abord que l’architecture “our w/ SC” améliore les performances pour une profondeur de 6 par rapport à une profondeur de 3. Néanmoins, l’architecture “our Decoder” est dans tous les cas meilleure, avec des performances assez stables sur les paramètres étudiées. Enfin, le DD a de meilleures performances en augmentant la profondeur jusqu’à 5, mais devient plus instable pour une profondeur de 6. L’augmentation du nombre de canaux améliore légèrement

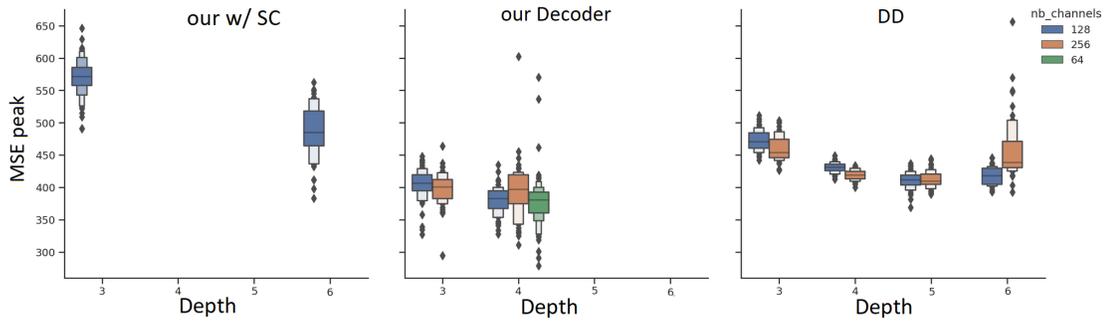


FIGURE 3.26 – Boîtes à moustaches de la distribution du MSE peak pour différentes valeurs de profondeur et nombre de canaux pour les architectures “our w/ SC” (*gauche*), “our Decoder” (*milieu*) et DD (*droite*).

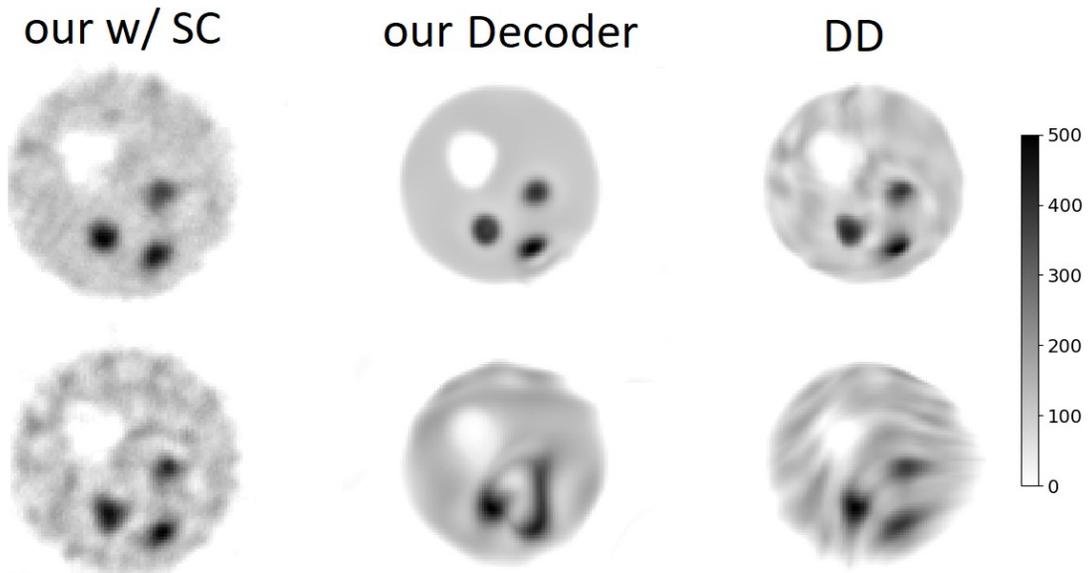


FIGURE 3.27 – Images obtenues pour les trois architectures étudiées pour le débruitage DIP. La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour chaque architecture.

les métriques, surtout pour les architectures les moins profondes. Les performances se situent entre l’architecture “our w/ SC” et “our Decoder”.

La figure 3.27 montre les images obtenues pour les trois architectures étudiées, correspondant aux points le plus bas et le plus haut des boîtes à moustaches de la figure 3.26. L’architecture “our Decoder” produit des images de bonne qualité, qui ont moins de distorsions par rapport à la GT que le DD. L’architecture “our w/ SC” produit quant à elle des images qui paraissent plus bruitées que les deux autres architectures.

3.5.6.3.3 Discussion L’architecture “our w/ SC” est la moins performante parmi les trois architectures étudiées. Les différences avec les deux autres architectures sont la présence d’un encodeur et des SC de l’encodeur au décodeur. Dans la suite, nous allons ajouter une architecture sans SC pour comparaison.

En outre, la meilleure MSE obtenue par le DD est de plus en plus en basse au fur et à mesure que l’on complexifie le réseau. La seule exception est l’architecture d’une profondeur de 6 et avec 256 canaux, où certaines images sont très mal débruitées. Cela s’explique par le fait que l’image en entrée est alors de très petite taille (2×2 pixels), pouvant donner peu de possibilité au DD de produire une image de bonne qualité.

3.5.6.4 Ajout de couches de contrôle spectral

Dans cette partie, nous avons réutilisé des couches de réseau de neurones introduites dans [81] pour permettre de contrôler le biais spectral de l’architecture utilisée pour le DIP :

- la première est la couche de normalisation de Lipchitz. La norme spectrale d’un noyau d’une couche convolutionnelle est saturée à une valeur λ sélectionnée manuellement. Cela a pour effet de borner la transformée de Fourier de la couche convolutionnelle en particulier pour les hautes fréquences. Les résultats dans [81] montrent un pic de PSNR qui reste stable au cours des itérations du DIP grâce à une valeur de λ bien choisie ($\lambda = 2$ dans leurs expériences). Cette couche est utilisée en remplacement de la couche de normalisation par lots,
- la deuxième est la couche d’interpolation gaussienne. Cette couche permet de contrôler le caractère lisse par morceau de l’image de sortie du DIP, qui induit une reconstruction plus ou moins rapide des hautes fréquences. Pour cela, cette couche effectue la convolution d’une couche convolution transposée (servant à réaliser un sur-échantillonnage en ajoutant des zéros entre les pixels de l’entrée de la couche) avec un filtre gaussien de variance σ^2 . Les résultats dans [81] montrent un pic de PSNR amoindri si σ est trop faible, et ralentit la convergence si σ est trop fort. Une valeur intermédiaire ($\sigma=0.5$ dans [81]) permet d’atteindre un pic PSNR stable au cours des itérations. Cette valeur permet d’obtenir de bonnes performances associées à une bonne vitesse de convergence vers le pic PSNR.

Ces deux couches peuvent être utilisées indépendamment.

3.5.6.4.1 Expériences Dans cette étude, nous avons poursuivi la comparaison des trois architectures précédentes (“our w/ SC”, “our Decoder” et DD) et ajouté l’architecture “our w/o SC”. Celle-ci est identique à l’architecture “our w/ SC” mais où les Sauts de Connexion (SC) ont été retirées. Nous avons fait varier la profondeur et le nombre de canaux comme dans l’expérience précédente.

De plus, nous avons aussi modifié les architectures “our Decoder” et DD en ajoutant les couches de normalisation de Lipschitz et d’interpolation gaussienne. Nous avons nommé ces architectures “our Decoder LG” et “DD LG”. Quatre valeurs de λ ont été utilisées ($\lambda \in \{0; 1; 1, 4; 2\}$). Nous parlerons de “contrôle LG” et ajouterons le suffixe “LG” pour les architectures utilisant ces deux couches.

Nous avons utilisé le fantôme cylindrique 3 dans le cadre de simulations ${}^{90}\text{Y}$. Les données sont reconstruites avec l’algorithme BSREM+RDP avec les mêmes paramètres que dans la partie 2.7.2.2.

Les images correspondant aux meilleures et aux moins bonnes configurations (en termes de MSE peak) pour chaque architecture ont été comparées.

3.5.6.4.2 Résultats La figure 3.28 présente les meilleure et moins bonne images obtenues pour chaque architecture. L’ajout du contrôle LG pour les architectures “our Decoder LG” et “DD LG” a amélioré la définition des tumeurs, en particulier dans le cas de la moins bonne image. Cependant, l’architecture “our Decoder LG” souffre d’artefacts du damier aussi bien dans la meilleure image obtenue que pour la moins bonne. Ces artefacts sont très marqués dans le cas de la moins bonne image. Pour l’architecture “DD LG”, l’ajout du contrôle LG a permis de diminuer les distorsions observées pour l’architecture DD, au prix d’une augmentation du niveau de bruit. De plus, les images obtenues par l’architecture “our w/ SC” semblent plus bruitées que celles obtenues par l’architecture “our w/o SC”. Enfin, l’architecture “our Decoder” propose des images similaires à l’architecture “our w/o SC”, bien que les tumeurs peuvent être moins bien définies dans le cas du décodeur seul.

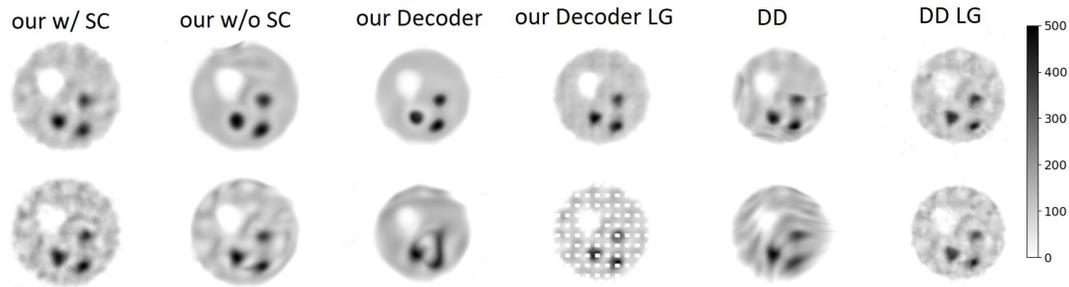


FIGURE 3.28 – Images obtenues pour les quatre architectures étudiées pour le débruitage DIP, avec ou sans contrôle LG pour les architectures avec uniquement un décodeur. La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour chaque architecture.

3.5.6.4.3 Discussion L’ajout du contrôle LG permet d’améliorer la définition de régions proches, mais au prix d’artefacts pour l’architecture “our Decoder LG” venant de la couche de convolution transposée, ou d’un plus fort niveau de bruit dans le cas du “DD LG”. Ces résultats sont différents de ceux obtenus dans [81] avec $\lambda = 2$ car pour toutes les valeurs de λ essayées dans notre expérience, les réseaux DIP finissent par reproduire l’image étiquette (la MSE n’atteint pas de plateau). Les différences d’architecture entre celles proposées par [81] et celles étudiées ici devraient être investiguées. Ainsi, l’ajout du contrôle LG n’améliore pas la qualité des images produites avec les architectures de type decoder utilisées.

En outre, la présence des SC avec entrée aléatoire semble apporter du bruit aux images de sortie du DIP, alors que l’encodeur semble améliorer leur qualité en réalisant une corrélation des pixels de l’image d’entrée. En définitive, la meilleure architecture obtenue après ces deux expériences et avec une entrée aléatoire est l’architecture “our w/o SC” que nous conserverons par la suite.

3.5.6.5 Ajouter du bruit à l’entrée ou à l’espace latent du réseau DIP

[18; 92] ont eu l’idée d’ajouter un bruit supplémentaire, différent à chaque itération du DIP, à l’image d’entrée aléatoire. Les auteurs dans [18] ont notamment observé que l’ajout d’un bruit gaussien à l’image d’entrée permet d’augmenter le pic PSNR et surtout permet de maintenir le pic pendant plus d’itérations.

3.5.6.5.1 Expériences Nous avons repris cette idée et avons considéré deux cas différents pour l’architecture “our w/o SC” :

- Cas v_0 : ajout d’un bruit uniforme $\mathcal{U}(0, \sigma_p)$ à l’entrée du DIP,

- Cas v_1 : ajout d'un bruit uniforme $\mathcal{U}(0, \sigma_p)$ au niveau de l'espace latent du DIP (entre l'encodeur et le décodeur de l'architecture).

Nous avons utilisé le fantôme cylindrique 3 dans le cadre de simulations ^{90}Y . Les données sont reconstruites avec l'algorithme BSREM+RDP avec les mêmes paramètres que dans la partie 2.7.2.2.

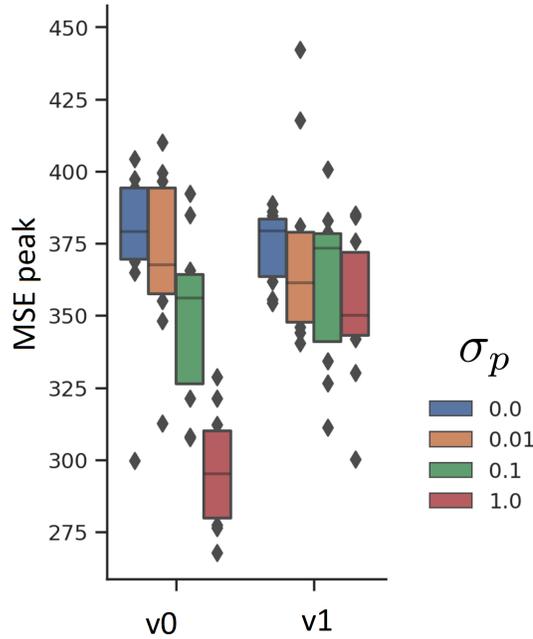


FIGURE 3.29 – Boîtes à moustaches pour les deux expériences avec injection de bruit dans l'image d'entrée (v_0) ou dans l'espace latent (v_1), pour différentes valeurs de σ_p .

3.5.6.5.2 Résultats La figure 3.29 présente les boîtes à moustaches obtenues dans les deux cas pour différentes valeurs de σ_p . Nous remarquons tout d'abord que pour les deux cas, l'ajout d'un bruit uniforme ($\sigma_p > 0$) permet d'améliorer la MSE des boîtes à moustaches. De plus, plus le niveau de bruit injecté dans l'image d'entrée est élevé, meilleure est la MSE. Cette différence est plus prononcée pour le cas v_0 qui obtient une meilleure MSE que le cas v_1 .

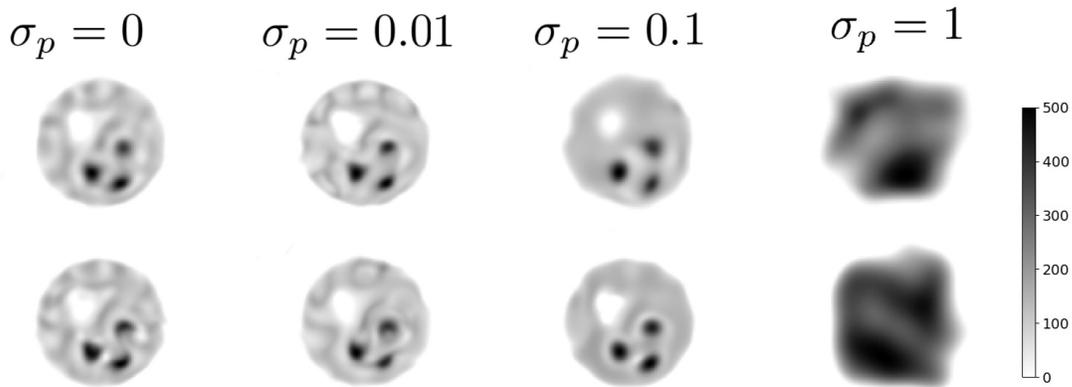


FIGURE 3.30 – Images obtenues pour l'expérience d'injection de bruit en entrée du réseau DIP, dans le cas v_0 . La première ligne (respectivement la deuxième ligne) montre les meilleures (respectivement les moins bonnes) images obtenues pour différentes valeurs de σ_p .

Les meilleures et moins bonnes images en termes de MSE peak sont reportées dans la figure 3.30 dans le cas v_0 , pour les valeurs de σ_p présentées dans la figure 3.29. Les images apparaissent de plus en plus lisses au fur et à mesure que σ_p augmente. En revanche, les images obtenues pour $\sigma_p = 1$ sont complètement inutilisables.

3.5.6.5.3 Discussion Les valeurs $\sigma_p \geq 1$ définissant le niveau de bruit injecté dans l'image d'entrée ont conduit à des images inutilisables. Les images deviennent alors beaucoup trop lisses et déformées. L'injection d'un bruit doit être d'un niveau moins important que le bruit dans l'image d'entrée (pour $\sigma_p < 1$) pour obtenir de bonnes performances qualitatives.

En outre, les images sont quantitativement meilleures avec injection de bruit en entrée du réseau DIP que sans injection de bruit. Cette observation est conjointe à des images de plus en plus lisses en augmentant le niveau de bruit injecté (cf. figure 3.30). Nous supposons que le bruit dans l'image étiquette est plus sensible aux changements des paramètres d'entrée du réseau, tandis que les informations représentant l'image naturelle dans l'image étiquette sont moins sensibles à ces changements. Ainsi, en incorporant de petites variations aléatoires au bruit d'entrée à chaque itération, tout en préservant la structure globale de l'image d'entrée d'origine, le réseau capturera en priorité l'image naturelle plutôt que le bruit dans l'image étiquette.

Remarque : l'ajout d'un bruit gaussien $\mathcal{N}(0, \sigma_p^2)$ a aussi été considéré dans les expériences et a donné des résultats similaires à l'ajout d'un bruit uniforme.

3.5.6.6 Architecture utilisant les transformateurs

Pour cette dernière expérience, nous avons voulu étudier plus en profondeur comment encoder l'information présente dans l'image d'entrée du réseau DIP puis la décoder. Nous nous sommes ainsi intéressés à des types d'architectures différents du U-net, et notamment les transformateurs. Nous proposons ici une introduction au concept des transformateurs et ne traiterons pas en détail des mathématiques concernant les mécanismes d'auto attention et d'attention multi-tête.

Les transformateurs ont été initialement proposés par [94] dans le domaine du traitement du langage et appliqués dans le monde du traitement d'images pour la première fois dans [28]. Ce type de réseau de neurones permet de traiter une séquence composée de tokens (ensemble de mots dans un texte ou ensembles de patches dans une image) dans son ensemble contrairement à des modèles de réseaux de neurones récurrents utilisés précédemment. En effet, pour traiter un nouveau token $p + 1$ d'une séquence, un réseau de neurones récurrent utilise ce token, ainsi que la p -ème sortie calculée précédemment, ce qui nécessite de traiter les tokens dans l'ordre et un par un.

Ainsi, les transformateurs sont devenus les modèles à l'état de l'art dans beaucoup de tâches impliquant des réseaux de neurones supervisés (traduction automatique, analyse de sentiment, génération d'images, etc.). Par exemple, dans le domaine du traitement d'images, les réseaux de neurones convolutionnels prennent en compte une dépendance entre pixels limitée spatialement, déterminée par la taille des noyaux de convolution et par la profondeur du réseau de neurones. Les transformateurs quant à eux permettent de modéliser des relations entre pixels plus éloignés grâce au traitement en parallèle de toute une séquence. Un encodage de position est utilisé pour leur permettre de retenir l'ordre des tokens dans une séquence. Enfin, les transformateurs s'appuient sur une nouvelle couche introduite dans [94] appelée auto attention, et reprise pour les transformateurs en imagerie [28]. Cette couche permet de déterminer un score, appelé attention, entre chaque patch d'une même image. Différents scores peuvent être calculés entre différents patches d'une image par l'utilisation de couche d'attention dit multi-tête. Celle-ci permet d'augmenter les capacités du réseau à modéliser des liens complexes entre les patches d'une image.

Une des clés du succès des transformateurs dans un contexte supervisé est de pouvoir être entraînés sur un très grand ensemble de données, ce qui demande plus de ressources informatiques que pour des réseaux de neurones récurrents ou convolutionnels [28]. Cependant, dans le cas du DIP, le traitement est non supervisé et utilise un seul jeu de données. De plus, les auteurs dans [19] expliquent que les réseaux de neurones utilisant des couches convolutionnelles induisent un biais spectral, qui est au fondement du DIP. Dans cette partie, nous souhaitons étudier le

comportement d'un transformateur et étudier sa capacité à induire ou non un biais spectral sur l'image de sortie.

3.5.6.6.1 Expériences Nous nous plaçons dans le cadre où une image anatomique est utilisée en entrée du réseau DIP. Nous voulons alors étudier la capacité de différentes architectures à encoder cette image d'entrée pour aider le débruitage DIP. Nous avons utilisé le fantôme cérébral du chapitre 2 (cf. figure 2.5) auquel a été retiré la tumeur des images TEP et anatomique.

Pour commencer, nous avons étudié l'effet du nombre de SC de 3 à 0 pour l'architecture "our w/ SC". Cette architecture a été comparée à deux autres s'appuyant sur les transformateurs originellement proposées pour le traitement d'images IRM :

- Swin Unet [14]. Cette architecture est composée d'un encodeur et d'un décodeur, chacun composé de blocs swin transformateur. Des SC sont utilisées pour passer de l'information entre l'encodeur et le décodeur directement,
- Swin UNETR [48]. Cette architecture est similaire à l'architecture Swin Unet, étant formée d'un encodeur et d'un décodeur. Cependant, seul l'encodeur est composé de blocs swin transformateur, alors que le décodeur est composé de couches convolutionnelles. Des SC sont aussi utilisées, mais où l'information de l'encodeur passe d'abord par une couche convolutionnelle avant d'être utilisée par le décodeur.

Un bloc swin transformateur est composé :

- d'une couche de normalisation (normalization layer en anglais, qui appartient à la même famille que la couche de normalisation par lots [105]),
- une couche d'attention multi-tête,
- une nouvelle couche de normalisation,
- un perceptron multi-couche. Le perceptron est au fondement des réseaux de neurones, où une entrée subit une transformation affine par les poids du réseau, puis passe à travers une fonction d'activation. Plusieurs couches sont alignées pour former le perceptron multi-couche.

Pour ces deux architectures, nous avons fait varier le nombre de têtes (3 à 6) et la profondeur du réseau (2 à 4). Toutes deux ont été comparées à l'architecture convolutionnelle "our w/ SC".

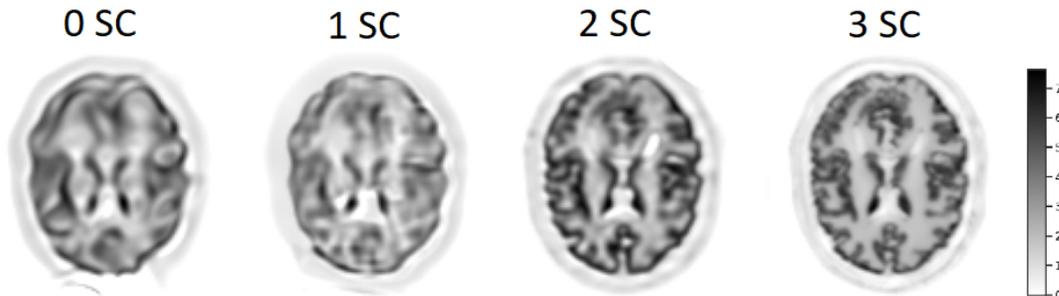


FIGURE 3.31 – Images correspondant au minimum de MSE obtenu pour le fantôme cérébral FDG avec l'architecture "our w/ SC" utilisée en tâche de débruitage DIP, pour différents nombres de SC.

3.5.6.6.2 Résultats La figure 3.31 montrent les images obtenues en faisant varier le nombre de SC dans l'architecture "our w/ SC". Les détails anatomiques sont de plus en plus présents avec l'augmentation du nombre de SC, notamment à partir de 2 SC. La figure 3.32 montre les boîtes à moustaches en fonction du nombre de SC. La MSE est toujours améliorée en augmentant le nombre de SC. L'architecture "our w/ SC" avec 3 SC obtient ainsi les meilleurs résultats.

Les meilleures images produites par les deux architectures utilisant des transformateurs sont comparées à l'architecture "our w/ SC" avec 3 SC dans la figure 3.33. L'image obtenue avec le Swin Unet présente moins de détails que les deux autres méthodes. L'image a toutefois été améliorée par rapport à l'image étiquette. Quant au Swin UNETR, l'image produite présente de

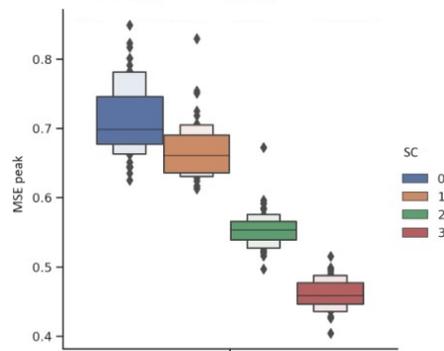


FIGURE 3.32 – Boîtes à moustaches obtenues avec l’architecture “our w/ SC” utilisée en tâche de débruitage DIP pour différents nombres de SC.

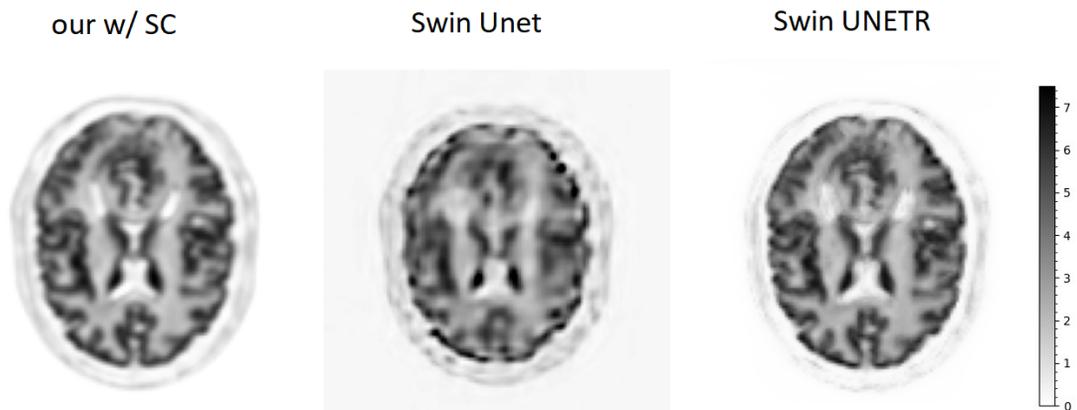


FIGURE 3.33 – Images correspondant au minimum de MSE obtenue pour le fantôme cérébral FDG avec les architectures “our w/ SC”, Swin Unet et Swin UNETR utilisées en tâche de débruitage DIP.

nombreux détails, avec des zones mieux définies que dans l'image produite par le Swin Unet. La meilleure image obtenue avec Swin UNETR est assez similaire à celle produite par l'architecture "our w/ SC", bien que le crâne ait un peu disparu et que des artefacts soient présents, notamment dans les ventricules latéraux.

3.5.6.6.3 Discussion Tout d'abord, les SC ont une utilité différente lorsqu'une image anatomique est utilisée en entrée du DIP plutôt qu'une image aléatoire. En effet, contrairement à une entrée aléatoire qui apporte du bruit à travers les SC, l'image anatomique apporte des détails qui améliorent la qualité de l'image débruitée ainsi que la MSE peak. La meilleure architecture de type "our w/ SC", dans le cas d'une image anatomique utilisée en entrée de réseau, est celle qui utilise toutes les SC.

De plus, cette architecture a été comparée qualitativement aux architectures utilisant des transformateurs. D'une part, le Swin Unet est l'architecture la moins performante : le minimum de MSE est obtenue pour une image certes débruitée, mais ne contenant pas autant de détails qu'avec les autres architectures. D'autre part, le Swin UNETR décode l'information issue de l'encodeur grâce à des couches convolutionnelles. Les détails sont beaucoup mieux récupérés dans l'image de sortie par rapport au Swin Unet, montrant la supériorité des couches convolutionnelles dans le décodeur. De plus, l'encodage de l'image anatomique par transformateur ne semble pas être bénéfique par rapport à l'encodeur avec couches convolutionnelles. D'autres architectures de type autoencodeur utilisant un encodeur avec transformateur pourraient être étudiées pour renforcer cette conclusion [79].

3.5.6.7 Discussion générale sur l'architecture du DIP en fonction de l'image d'entrée

L'image d'entrée du réseau DIP peut être choisie de façon aléatoire ou en utilisant une image anatomique.

Pour une image aléatoire, les SC sont sources de bruit dans l'image de sortie, ce qui amène à considérer une architecture autoencodeur convolutionnelle sans SC ou alors uniquement de type décodeur. L'ajout d'un bruit d'un niveau légèrement inférieur au bruit de l'image d'entrée à chaque itération permet de rendre les images de sortie du réseau DIP plus lisses et améliore le minimum de MSE atteint. Le bruit améliore le minimum de MSE de façon plus importante lorsqu'il est injecté au niveau de l'entrée plutôt qu'au niveau l'espace latent.

Dans le cas d'un image anatomique utilisée en entrée du réseau DIP, les SC permettent d'améliorer les détails dans l'image de sortie du DIP ainsi que le minimum de MSE atteint. De plus, les couches convolutionnelles utilisées dans le décodeur induisent un biais spectral plus fort que les blocs swin transformateur utilisés. L'utilisation de blocs swin transformateurs au niveau de l'encodeur n'a pas amélioré les détails dans l'image de sortie du réseau DIP.

Les résultats obtenus dans ces expériences montrent que le choix de l'architecture du DIP dépend de l'image d'entrée utilisée. Dans une prochaine étude, il serait important de comparer les meilleurs paramétrages obtenus avec entrée aléatoire et entrée anatomique sur un même fantôme. Notons aussi que l'utilisation d'une architecture avec uniquement un décodeur ne peut pas incorporer directement une information anatomique. Un deuxième réseau de neurones peut être utilisé en parallèle du DIP pour encoder l'image anatomique dans l'espace latent du DIP et être utilisé en entrée de celui-ci, comme proposé par [68]. Pour le DNA dans le chapitre 2, nous avons utilisé l'architecture convolutionnelle donnant les meilleurs résultats avec image anatomique lors de ces expériences. L'utilisation d'une image anatomique dans le cas FDG a permis d'améliorer visuellement les détails dans le cerveau (cf. partie 2.5) pour un niveau de bruit faible par rapport aux autres méthodes utilisées. Dans le cas ^{90}Y , le bruit dans l'image reconstruite est plus important. Il serait alors intéressant de considérer une image aléatoire en entrée, combinée à une architecture sans SC et l'injection d'un bruit aléatoire à chaque itération externe dans le but de réduire la quantité de bruit dans l'image reconstruite par le DNA.

Dans le dernier chapitre, nous voulons étudier l'influence des SC au sein de la reconstruction en combinaison avec l'image d'entrée utilisée. En effet, les SC n'ont pas le même intérêt en fonction de l'image d'entrée utilisée d'après les expériences menées pour le débruitage DIP. Ce dernier chapitre va élargir l'étude à la reconstruction TEP.

3.6 Choix du nombre d'itérations des blocs itératifs du DNA

Le dernier type d'hyperparamètres à étudier et le nombre d'itération de chaque bloc itératif du DNA (cf. figure 2.2). Dans chacune des parties suivantes, nous allons voir l'influence du nombre d'itérations de chacun de ces blocs. Le lecteur est invité à se référer au diagramme 2.2 et aux définitions données pour chaque nombre d'itérations dans la partie 2.3.2.

3.6.1 Influence du nombre d'itérations du DIP au sein de la reconstruction

3.6.1.1 Expériences

Dans un premier temps, nous avons étudié l'influence du nombre d'itérations du DIP sur les propriétés des images reconstruites. Le DNA a été utilisé avec le même paramétrage que dans la partie 2.7.2.2, en faisant varier le nombre d'itérations du DIP (30, 100, 300 itérations). 700 itérations externes ont été utilisées pour chacun de ces paramétrages.

Dans un second temps, nous avons voulu voir si ce choix du nombre d'itérations du DIP pendant les itérations externes du DNA pouvait être effectué automatiquement grâce à l'algorithme EMV, de la même manière que nous l'utilisons pour l'initialisation du DNA.

Nous avons utilisé les simulations ^{90}Y du fantôme cylindrique 2.

3.6.1.2 Résultats

La figure 3.34 montre les courbes de compromis obtenues pour trois valeurs différentes du nombre d'itérations du DIP. Nous observons tout d'abord une évolution de la rugosité de l'image plus rapide du DNA avec un nombre croissant d'itérations du DIP. Un comportement différent pour 30 itérations du DIP par rapport aux deux autres paramétrages est observé. La récupération d'activité dans la région "TEP seule" est d'abord plus faible, mais associée à une faible rugosité de l'image. Par la suite, la courbe continue de croître avec une augmentation de la rugosité de l'image, contrairement aux deux autres courbes (100 et 300 itérations) qui changent de chemin juste après leur initialisation. Globalement, pour une rugosité fixe, la récupération d'activité est meilleure pour avec 30 itérations du DIP qu'avec 100 ou 300. Le SSIM est aussi légèrement meilleur pour 30 itérations du DIP. Enfin, le biais diminue beaucoup plus vite pour une faible rugosité avec 30 itérations qu'avec 100 ou 300 itérations du DIP.

La figure 3.35 représente un graphique de la valeur de l'EMV en fonction du nombre d'itérations du DIP, pour l'initialisation du DNA (itération -1) et les 5 premières itérations externes (numérotées de 0 à 4). La courbe de l'initialisation du DNA se détache des autres courbes, avec une EMV globalement la plus élevée et s'arrêtant de décroître autour de 250 itérations avant de subir de légères oscillations. Pour les autres itérations externes, les courbes sont aussi décroissantes lors des premières itérations du DIP mais atteignent un minimum local plus rapidement puis subissent des d'importantes oscillations. Le minimum global atteint se situe entre 100 et 300 itérations.

Enfin, la figure 3.36 présente deux images reconstruites par le DNA avec utilisation de l'algorithme EMV à chaque itération externe, pour les itérations externes 46 et 47. Les images sont très différentes : l'une est proche de la GT, l'autre s'approche davantage de l'image utilisée en entrée du DIP.

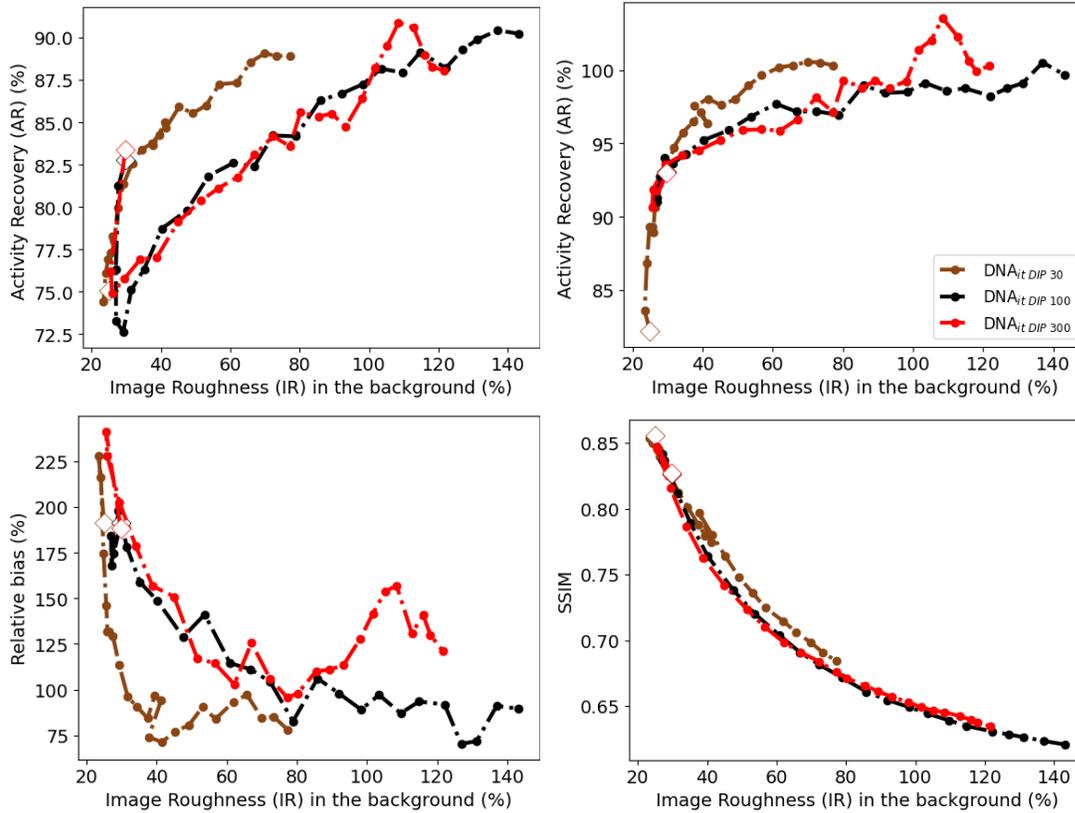


FIGURE 3.34 – Courbes de compromis pour 15 répliqués pour le fantôme cylindrique 2, dans la ROI de la tumeur “TEP seule” (*haut gauche*), la ROI de la tumeur “TEP match” (*haut droite*), la ROI de la zone froide (*bas gauche*) et le SSIM dans l’ensemble du fantôme (*bas droite*) en fonction de la rugosité dans le fond du fantôme, pour l’algorithme DNA utilisant 30, 100 ou 300 itérations du DIP. Les diamants blancs montrent les métriques à la première itération externe.

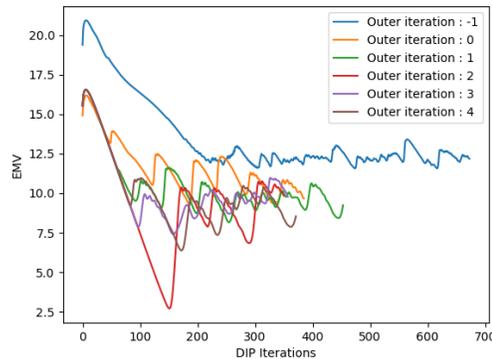


FIGURE 3.35 – Valeur de l’EMV en fonction des itérations du DIP lors d’une reconstruction avec l’algorithme DNA utilisant l’EMV à chaque itération externe. Chaque courbe correspond à l’EMV calculée pour une itération externe donnée. L’itération -1 correspond à l’initialisation du DNA. Les courbes s’arrêtent 200 itérations après l’itération choisie par l’algorithme EMV.

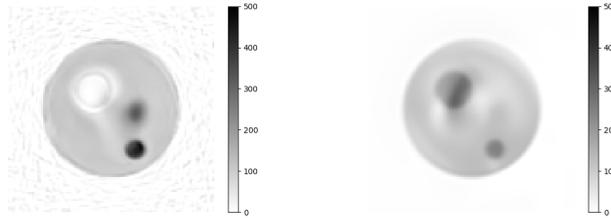


FIGURE 3.36 – Images reconstruites par le DNA avec utilisation de l’algorithme EMV à chaque itération du DIP au sein de la reconstruction. L’image de gauche montre l’image reconstruite par le DNA après 46 itérations externes, celle de droite après 47 itérations externes.

3.6.1.3 Discussion

Le nombre d’itérations du DIP au sein du DNA influence le débruitage de l’image reconstruite par l’étape ADMM-Reg : plus le nombre d’itérations du DIP est élevé, plus le DIP réintroduit des hautes fréquences ce qui se traduit par une évolution rapide de la rugosité au cours des itérations externes du DNA.

L’algorithme EMV utilisé à chaque itération externe pourrait présenter un intérêt pratique pour ne pas avoir à fixer le nombre d’itérations du DIP manuellement. La figure 3.35 montre que l’itération choisie par l’algorithme EMV se situe toujours au delà de 100 itérations. Étant donné les résultats présentés dans la figure 3.34 montrant une nette supériorité du choix de 30 itérations par rapport à 100 ou 300 itérations du DIP, il semble inapproprié d’utiliser l’algorithme EMV pour déterminer le nombre d’itérations du DIP à l’intérieur du DNA. Cette conclusion est renforcée par les résultats présentés dans la figure 3.36 montrant deux itérations externes successives très différentes, reflétant ainsi un comportement chaotique du DNA avec utilisation de l’algorithme EMV au sein de la reconstruction. Nous supposons que le DIP change de minimum local ce qui refait démarrer le débruitage à partir de zéro.

3.6.1.4 Conclusion

30 itérations du DIP au sein de la reconstruction ont donné de meilleurs résultats que 100 et 300 itérations du DIP pour des simulations ^{90}Y . L’algorithme EMV utilisé au sein de la reconstruction avec le DNA ne permet pas de reconstruire une image de qualité. Dans le chapitre 2, nous avons utilisé 100 itérations du DIP pour le cas ^{90}Y et 200 pour le cas FDG.

3.6.2 Choix du nombre de sous-itérations internes

Plus le nombre de sous-itérations internes est élevé, plus rapidement l’image x^k se rapproche du maximum de vraisemblance à l’itération externe k du DNA, au prix d’une augmentation du bruit plus rapide. Comme expliqué dans la partie 2.3.2.1, ce paramètre a été fixé à 1 dans cette étude. C’est aussi le paramétrage utilisé par l’algorithme ADMM-Reg dans [59]. Seul le nombre d’itérations internes est varié comme expliqué dans la partie suivante.

3.6.3 Influence du nombre d’itérations internes

L’effet du nombre d’itérations internes a été étudié conjointement au paramètre de pénalité ADMM ρ_1 dans la partie 3.5.3. Pour un paramètre ρ_1 fixé, le bruit dans l’image reconstruite augmente avec le nombre d’itérations internes. Pour un paramètre ρ_1 faible, le nombre d’itérations internes ne doit pas être trop élevé pour éviter les artefacts de lignes présentés dans la partie 3.5.3.

En pratique, le nombre d’itérations internes doit être choisi petit pour éviter de possibles artefacts. Par exemple, dans la partie 2.4.2.2, 3 itérations internes ont été réalisées. Si les images n’évoluent pas beaucoup entre les itérations externes du DNA, un plus grand nombre peut être

choisi tant que des artefacts n'apparaissent pas. C'est ce que nous avons décidé dans la partie 2.4.3.

3.6.4 Choix du nombre d'itérations externes

Dans cette partie, nous avons étudié l'impact du choix du nombre d'itérations externes de DNA. Nous avons utilisé les simulations ^{90}Y du fantôme cylindrique 2 et le paramétrage présenté en partie 2.7.2.2 du DNA. Nous commençons par présenter des images reconstruites pour différents nombres d'itérations externes et discutons ensuite quelques perspectives pour un choix approprié de cet hyperparamètre. Ces perspectives sont valables pour les algorithmes DNA et DIPRecon.

3.6.4.1 Observations

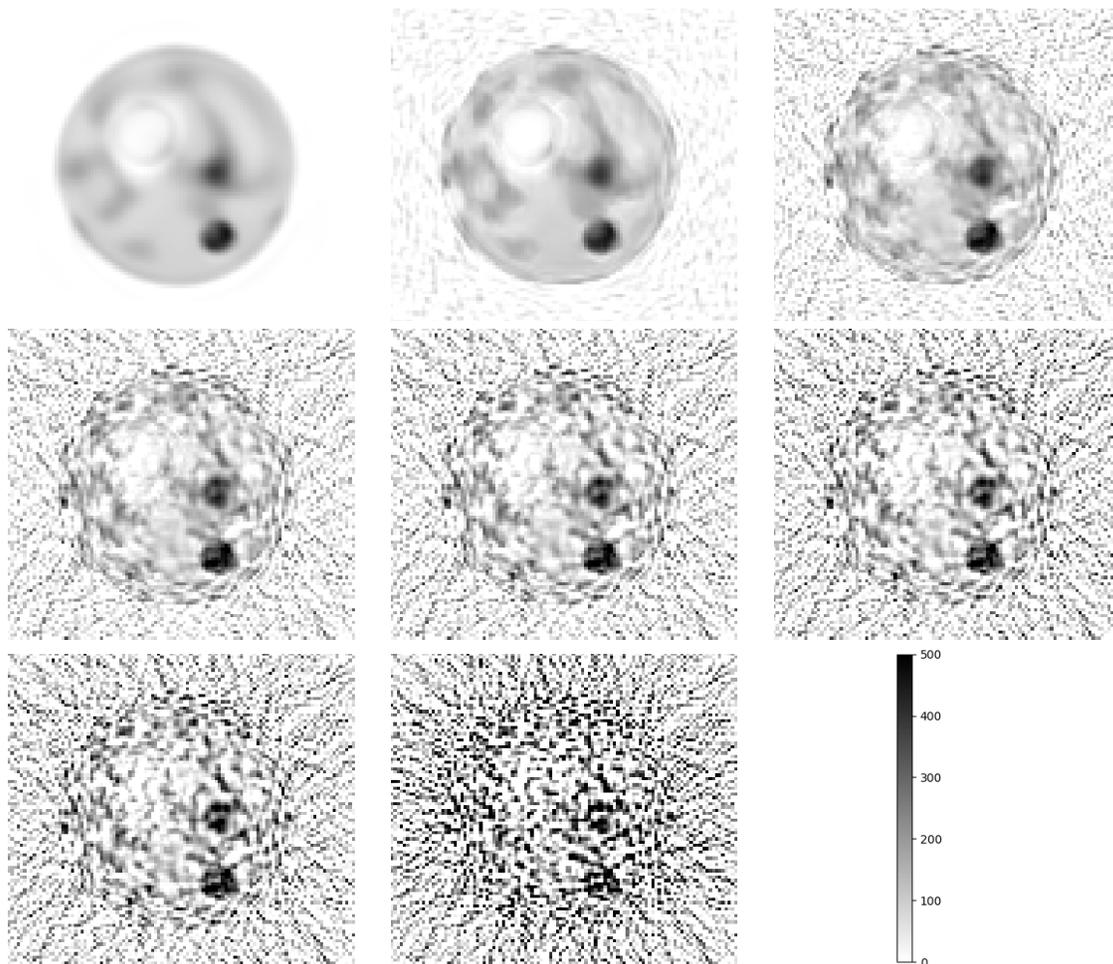


FIGURE 3.37 – Images reconstruites des simulations ^{90}Y du fantôme cylindrique 2 avec l'algorithme DNA (même paramétrage qu'en partie 2.7.2.2). L'initialisation DIP-BSREM du DNA est présentée en haut à gauche, puis chaque image correspond à un nombre d'itérations externes de 100 jusqu'à 600 et la dernière image est pour 1700.

La figure 3.37 montre les images reconstruites par le DNA toutes les 100 itérations externes jusqu'à 600, ainsi qu'une image reconstruite après 1700 itérations. Dans les premières itérations externes du DNA, l'image est assez lisse mais présente une activité dans les régions chaudes légèrement sous-estimée. Après plusieurs centaines d'itérations, l'image reconstruite devient extrêmement bruitée. Comme déjà discuté dans la partie 2.6, le DNA se comporte comme

ADMM-Reg à convergence et produit donc des images bruitées. Le nombre d'itérations externes est donc un paramètre capital pour obtenir une image reconstruite par le DNA à la fois qualitative et quantitative. Ces observations s'appliqueraient aussi au DIPRecon.

Le choix du nombre d'itérations externes dépend aussi du nombre d'itérations internes et du paramètre de pénalité ADMM ρ_1 . En effet, plus le nombre d'itérations internes est grand, moins il faut faire d'itérations externes pour atteindre le même niveau de bruit. De même, plus le paramètre de pénalité ADMM ρ_1 est faible, moins il faut d'itérations externes pour avoir le même niveau de bruit (cf. partie 3.5.3). Par exemple, dans le chapitre 2, le paramètre ρ_1 pour reconstruire des images avec le DNA a été choisi bien plus faible pour le cas 3D (cf. partie 2.4.3 par rapport au cas 2D FDG 2.4.2). Cela explique pourquoi, dans le chapitre 2, nous avons choisi de présenter des images 3D reconstruites avec le DNA pour moins d'itérations externes par rapport aux simulations 2D FDG (cf. partie 2.4.3).

3.6.4.2 Perspectives

Dans le chapitre 2, le nombre d'itérations externes pour présenter les images reconstruites par DNA et DIPRecon dans les figures 2.8 2.9 et 2.12 a été choisi manuellement. Pour pouvoir utiliser ces algorithmes en clinique, il faudrait définir une méthode pour déterminer le nombre d'itérations externe de façon automatique. Sinon, une autre solution serait de pallier le surapprentissage du DIP.

3.6.4.2.1 Extension de l'algorithme EMV Pour sélectionner le nombre d'itérations externes de façon automatique, une piste serait d'étendre l'intuition de l'algorithme EMV appliqué dans cette thèse à un débruitage DIP, à tout l'algorithme DNA. Plutôt que de regarder la variance des images produites par des itérations successives du DIP, nous pourrions étudier la variance des images générées à chaque itération externe du DNA. Cette perspective pourrait s'appliquer aussi à l'algorithme DIPRecon.

3.6.4.2.2 Pallier le surapprentissage du DIP dans le cadre d'une intégration à la reconstruction TEP Une autre piste serait d'opter pour une solution remédiant au problème de surapprentissage du DIP. On peut supposer que dans ce cas, le DNA aurait en pratique un comportement convergent. Il suffirait alors d'utiliser un critère d'arrêt à convergence pour déterminer le nombre d'itérations externes du DNA et du DIPRecon.

De nombreuses études ont cherché à améliorer les capacités du DIP, en essayant d'atténuer cet effet sous-jacent de surapprentissage. Pour cela :

- une régularisation explicite peut être ajoutée [46; 53; 81; 88; 100] à celle du DIP. Dans le cas du DNA et du DIPRecon, cela reviendrait à ajouter explicitement un terme de régularisation à la fonction de vraisemblance (équations (2.14) et (2.3)). Cependant, ces méthodes reposent sur l'ajustement d'un paramètre de pénalité comme pour les algorithmes PML (cf. figure 3 dans [46], [99]). De plus, nous émettons quelques réserves sur ce type d'algorithme. En effet, lors de l'utilisation d'un réseau de neurones avec beaucoup plus de paramètres que de voxels dans l'image, comme c'est le cas lors de l'utilisation d'une architecture de type U-Net, nous pensons que le réseau peut apprendre à reproduire n'importe quelle image étiquette, même du bruit. Ainsi, sans régularisation explicite et à convergence, le DIP reproduit l'image étiquette et n'est finalement plus utile. De même, avec régularisation explicite et à convergence, le DIP devrait réussir à reproduire l'image correspondant à l'optimum théorique après ajout de la régularisation. Une justification théorique ou des expériences doivent être menées pour valider ou non cette hypothèse.
- l'estimation bayésienne a également été étudiée pour les images naturelles [22], et appliquée à la reconstruction TEP [17; 81]. Ce type de méthode permet d'obtenir une estimation d'incertitudes sur l'image reconstruite. En outre, les auteurs dans [17] ont observé dans leurs expériences que le surapprentissage du DIP était moins prononcé.
- [24] a utilisé une fonction de coût pour optimiser le DIP s'appuyant sur l'estimateur SURE (Stein Unbiased Risk Estimator), permettant à la sortie du DIP de rester proche de l'image GT inconnue. Cette méthode a été testée pour le débruitage d'image TEP

avec le DIP, et prévient son surapprentissage pour un hyperparamètre de l'estimateur SURE bien choisi,

- [58] proposer de modéliser un bruit parcimonieux directement dans la fonction de coût du réseau DIP. De cette façon, le bruit n'est pas récupéré dans l'image mais à travers un terme de la fonction de coût à optimiser. Toutefois, cette méthode suppose que le bruit est parcimonieux et connu, ce qui n'est pas le cas dans l'image TEP. De plus, cette méthode nécessite l'ajustement d'un hyperparamètre pour contrôler le niveau de bruit de l'image de sortie du DIP,
- l'utilisation de l'algorithme EMV pourrait être une idée, mais a donné de mauvais résultats quand il est utilisé au sein de la reconstruction (cf. partie 3.6.1).

Ces méthodes de la littérature demandent plus d'investigation pour être appliquées au sein du DNA et du DIPRecon. De plus, celles-ci nécessitent toutes le réglage d'un nouvel hyperparamètre à la place du nombre d'itérations externes. Pour une étude plus exhaustive et élargie au delà du cadre de la TEP, nous invitons le lecteur à lire le papier et notamment l'introduction du papier décrivant les méthodes WMV et EMV [99].

3.6.4.3 Conclusion

Le nombre d'itérations externes est important à choisir car les images produites par le DNA finissent par être bruitées à cause du surapprentissage du DIP. Ce nombre a été choisi manuellement dans le chapitre 2.

3.7 Discussion générale sur les hyperparamètres et conclusion

Pour tirer profit de l'algorithme DNA, de nombreux hyperparamètres doivent être choisis, affectant la vitesse de convergence de l'algorithme ou encore la qualité des images reconstruites. Ainsi, dans ce chapitre, nous avons investigué chaque hyperparamètre du DNA pour en comprendre l'influence. Un paramétrage idéal a pu être déterminé pour certains, tandis que d'autres sont intimement liés, alors que d'autres ont pu être réglés par des méthodes automatiques.

Le paramètre de pénalité ADMM ρ_2 tout comme le nombre d'itérations du DIP initial sont mis à jour de façon automatique respectivement grâce à la méthode des résidus relatifs et à l'algorithme EMV. D'une part, le paramètre ρ_2 est mis à jour de façon automatique par la méthode des résidus classiques dans [59]. La méthode des résidus relatifs a été utilisée ici, permettant un choix plus aisé des hyperparamètres associés à cette méthode par rapport à la méthode des résidus classiques. Celle-ci permet de gagner en temps de calcul ainsi que de ne pas ajouter d'hyperparamètres difficiles à régler par rapport au DIPRecon. D'autre part, le problème d'optimisation avec DIP étant non convexe, l'initialisation a une grande importance. L'algorithme EMV permet une très bonne initialisation à la fois quantitative et qualitative. A notre connaissance, c'est la première fois que cette méthode a été introduite pour arrêter le DIP de manière précoce en TEP.

Dans cette thèse, nous avons choisi d'initialiser l'algorithme DNA en partant d'une image de départ reconstruite avec un algorithme itératif, puis débruitée par le DIP. Nous avons constaté que l'image de départ BSREM a conduit à de très bonnes performances du DNA, en particulier dans les régions chaudes. Le DNA parvient à effectuer une réduction de biais dans les régions froides, au prix d'une augmentation du bruit dans l'image, de façon comparable aux algorithmes de réduction de biais APPGML et ADMM-Reg. L'algorithme ADMM-Reg pourrait aussi être utilisé pour reconstruire l'image de départ si on souhaite privilégier une réduction de biais dans les régions froides pour un niveau de bruit raisonnable plutôt que la récupération d'activité dans les régions chaudes.

Les voxels négatifs sont autorisés dans le DNA de façon mathématique par l'algorithme ADMM-Reg utilisé à l'intérieur du DNA. Néanmoins, l'algorithme APPGML, bien que n'ayant pas de justification mathématique, a été testé au sein de DNA. Il permet d'obtenir des résultats identiques sur un fantôme d' ^{90}Y , et une évolution plus rapide de l'algorithme dans le cas FDG. L'utilisation de l'algorithme APPGML à la place d'ADMM-Reg peut également provoquer des

artefacts donnant lieu à une divergence du DNA-APPGML si la borne A de APPGML est mal ajustée au problème. L'algorithme ADMM-Reg utilisé au sein du DNA a été préféré pour les expériences concernant l'étude des hyperparamètres pour sa dérivation mathématique et éviter le choix de la borne A de APPGML. L'algorithme DNA-APPGML sera utilisé dans le chapitre suivant pour son évolution plus rapide, permettant de limiter ainsi le temps de calcul.

L'optimisation du réseau DIP conduit à des images débruitées de bonne qualité lorsque l'optimiseur, le taux d'apprentissage, et l'architecture utilisés sont bien choisis. Nos expériences ont montré que l'optimiseur Adam permettait d'obtenir une meilleure rugosité de l'image par rapport à l'optimiseur L-BFGS. Pour que le biais spectral du réseau DIP soit présent, le taux d'apprentissage ne peut pas être choisi trop faible, ni trop élevé pour permettre de s'approcher au plus près de l'image idéalement débruitée. La valeur de 0,01 a été gardée après étude de la correspondance en bande de fréquences des images en sortie du DIP. La meilleure architecture du réseau DIP que nous avons obtenue pour le débruitage d'images TEP est un auto-encodeur avec couches convolutionnelles, utilisant des SC si une image anatomique est utilisée en entrée du réseau DIP, ou aucune SC si du bruit aléatoire est utilisé. Dans ce dernier cas, l'ajout de bruit supplémentaire à l'image d'entrée au cours des itérations peut conduire à réduire encore plus le bruit dans les images en sortie du réseau DIP. La mise à l'échelle utilisée pour l'image d'entrée et l'image étiquette n'a pas d'influence significative pour l'algorithme DNA. En revanche, à cause de la ReLU finale dans l'architecture utilisée par le DIPRecon, une mise à l'échelle de type normalisation au maximum (respectivement normalisation ou standardisation) résulte en un biais négatif (respectivement positif) dans les régions froides.

Enfin, les nombre d'itérations de chaque bloc du DNA doivent être sélectionnés avec attention. D'une part, le nombre d'itérations du DIP peut conduire à des diminutions brusques de la récupération d'activité au cours des itérations externes du DNA s'il est choisi trop élevé. Pour sélectionner ce nombre automatiquement, nous avons vu que l'utilisation de l'algorithme EMV au sein du DNA a conduit à un comportement chaotique. Cet algorithme semble être mieux adapté pour le débruitage DIP lors de l'initialisation du DNA, certainement car les poids du réseau sont initialisés aléatoirement, à la différence des poids réutilisés entre les itérations externes du DNA. D'autre part, le nombre d'itérations internes est intrinsèquement lié au paramètre de pénalité ADMM ρ_1 . Pour le DNA, le nombre d'itérations internes doit être choisi faible pour limiter les possibilités d'artefacts, en particulier si ρ_1 est choisi faible, et si l'acquisition est réalisée avec du FDG. Ce nombre peut être augmenté pour faire évoluer l'algorithme plus rapidement si des artefacts ne sont pas observés, en particulier dans une acquisition ^{90}Y , ou bien avec le DNA-APPGML.

Dans le dernier chapitre, nous avons souhaité nous intéresser à deux autres hyperparamètres plus en profondeur : les SC de l'architecture du DIP et l'entrée du réseau DIP. Ces paramètres ont été étudiés dans cette partie dans le cas d'un simple débruitage DIP, mais pas dans le cadre d'une reconstruction incluant le DIP. Nous désirons étudier aussi leur influence dans un contexte de discordances entre l'image anatomique et l'image TEP.

Chapitre 4

Influence de l'entrée du réseau DIP et des sauts de connexion avec utilisation d'information anatomique sur la reconstruction d'images TEP

4.1 Introduction

De nos jours, les scanners TEP cliniques sont quasi-systématiquement couplés à une modalité d'imagerie anatomique (TDM ou IRM). Les médecins regardent ainsi l'image anatomique en parallèle de leur analyse sur l'image TEP.

Étant donnée la meilleure résolution spatiale des images anatomiques par rapport aux images TEP,

Les images anatomiques sont caractérisées par une résolution spatiale (de l'ordre du millimètre) meilleure que les images TEP (de l'ordre de quelques millimètres). Plusieurs méthodes ont ainsi été proposées dans le but d'introduire de l'information anatomique lors de la reconstruction TEP afin d'améliorer la résolution spatiale :

- régularisation MRF avec des poids de Bowsher [78] (cf. partie 1.2.5.7.2.3),
- méthode à noyaux [98] : plutôt que de reconstruire directement une image x , on peut utiliser une matrice à noyaux K contenant de l'information a priori (par exemple anatomique) et chercher à reconstruire α tel que $x = K\alpha$,
- DIP avec entrée anatomique [25; 37] : un réseau de neurones non supervisé permettant de débruiter ou de restaurer une image sans entraînement préalable (cf. partie 2.2.2).

[37] obtient de meilleurs résultats en termes de récupération de contraste par rapport à la variance d'une zone uniforme avec l'algorithme DIPRecon en comparaison à la méthode à noyaux de [98]. De plus, nous avons vu que le DIP utilisé au sein du DNA incorporait l'information anatomique de façon plus naturelle par rapport à la méthode de Bowsher [78] (cf. partie 2.5).

Bien qu'il ait été démontré que le DIP conditionnel améliore les détails dans le cerveau [25; 37], il peut exister des disparités entre l'image anatomique et l'image TEP. Les auteurs dans [37; 44; 69] ont respectivement montré que le DIP conditionnel dans une tâche de débruitage des images TEP pouvait préserver une tumeur TEP absente de l'image IRM. Dans [38], une couche supplémentaire a été ajoutée après la structure en U-Net pour réaliser un changement d'espace comme les méthodes à noyaux, dans le but de pouvoir reconstruire des données TEP dynamiques. Dans ce cadre dynamique, cela a permis de mieux reconstruire les tumeurs TEP absentes de l'image IRM par rapport aux méthodes à noyaux. Les auteurs dans [68] ont tiré les mêmes conclusions avec leur méthode appliquée au débruitage d'images TEP, utilisant un décodeur pour

l'image TEP, alimenté par des informations IRM provenant d'un encodeur-décodeur spécifique à l'image IRM. Un type de méthode différent a également été proposé par [41], qui utilise un algorithme EM utilisant un noyau régularisé. L'image IRM a été utilisée pour définir la matrice du noyau, tandis que la pénalité présente un terme d'énergie fondée sur une image TEP pré-reconstruite. L'utilisation de cette image TEP a permis d'obtenir des bords plus définis et a favorisé les petits objets présents en TEP, mais manquants dans l'image IRM.

Cependant, ces études se sont principalement concentrées sur les tumeurs apparaissant en TEP, mais pas en IRM. D'autres discordances (appelées mismatches par la suite) entre l'image anatomique et l'image TEP peuvent être présentes, ce qui peut altérer la reconstruction. On peut les séparer en deux catégories :

- les mismatches de recalage : l'image TEP n'est pas bien alignée avec l'image anatomique. Cela arrive entre deux acquisitions ayant lieu sur des machines différentes, ou si le patient a bougé entre l'acquisition TEP et l'acquisition TDM ou IRM. Les performances de la reconstruction avec le DIP décroissent dans ce cas [69],
- les mismatches de contours de tumeurs : le contour d'une tumeur dans l'image TEP peut-être différent de celui dans l'image anatomique.

Dans ce travail, nous nous intéresserons seulement aux mismatches de contour. Toutes les études citées précédemment se sont concentrées sur les tumeurs TEP uniquement, ou sur les tumeurs ont des contours concordants entre l'image TEP et l'image anatomique. Dans ce chapitre, nous allons étudier d'autres types de mismatches possibles, avec des formes différentes entre l'image TEP et l'image anatomique, ainsi que des tumeurs ayant une information uniquement anatomique, et qui n'apparaissent donc pas sur l'image TEP.

Nous étudierons en premier lieu l'effet d'une reconstruction avec DIP conditionnel dans le cadre de multiples informations discordantes, par rapport au DIP avec bruit aléatoire en entrée. Nous proposerons ensuite deux méthodes, nommées "MR init" et "MR 5", ayant pour but de mitiger l'impact des mismatches entre les contours des tumeurs des images TEP et IRM. Les méthodes seront évaluées dans le cadre d'une simulation TEP d'une acquisition cérébrale FDG avec imagerie IRM T1 associée.

4.2 Simulations analytiques et paramètres de reconstruction généraux

4.2.1 Fantôme 2D cérébral et simulations

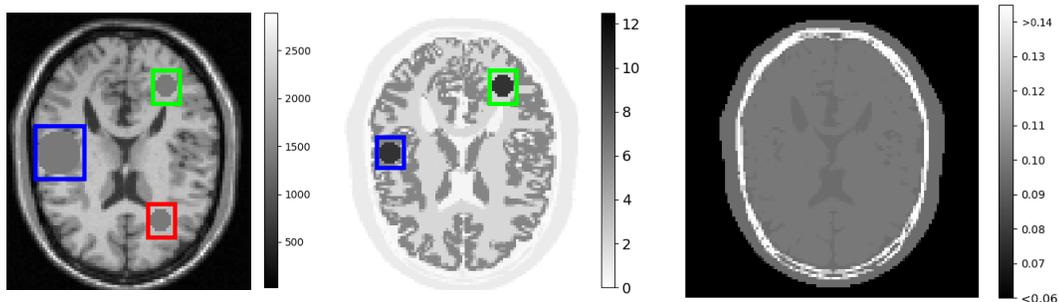


FIGURE 4.1 – Fantôme cérébral conçu pour l'étude des mismatches entre image TEP et image anatomique. À gauche, l'image anatomique ; au milieu, la vérité de base de la TEP ; à droite, l'image d'atténuation en cm^{-1} . Les cadres montrent les 3 types de tumeurs considérées : "TEP match" (vert), "TEP match" (bleu) et "IRM seule" (rouge).

Un fantôme cérébral 2D a été créé à partir de la base de données BrainWeb comme expliqué pour le fantôme cérébral FDG du chapitre 2 (cf. partie 2.4.2). Nous créons ainsi une image IRM T1 et une image TEP GT. Nous avons ajouté des tumeurs à ce fantôme comme présenté sur la figure 4.1 :

- une région tumorale avec concordance parfaite entre l’image IRM et la TEP (encadrée en vert dans la figure 4.1). La tumeur étant circulaire, on définit la ROI “TEP match” correspondant au cercle intérieur à la tumeur présente dans l’image TEP, à laquelle les voxels du bord de la tumeur ont été retirés,
- une région tumorale non-concordante entre l’image IRM et la TEP (encadrée en bleu dans la figure 4.1). La ROI correspondant au cercle intérieur de la tumeur TEP sera appelée “TEP match”,
- une tumeur apparaissant uniquement dans l’image IRM (encadrée en rouge dans la figure 4.1). La ROI correspondant au cercle intérieur de la tumeur IRM sera appelée “IRM seule”.

Une tumeur n’apparaissant que dans l’image TEP n’a pas été considérée dans cette étude car, normalement, une tumeur cérébrale présente sur l’image TEP est toujours présente sur l’image IRM.

Les valeurs SUV ont été fixées comme pour le fantôme cérébral FDG du chapitre 2 (cf. partie 2.4.2). L’image d’atténuation est la même façon que pour le fantôme du chapitre 2 (figure 4.1).

Une acquisition FDG a été simulée avec les mêmes paramètres que dans le chapitre 2 (partie 2.4.2), mais avec 10M de prompts. 40 réplicats statistiques ont été utilisés.

4.2.2 Paramètres de reconstruction

Les paramètres de reconstruction sont les mêmes que ceux utilisés dans le chapitre 2 (cf. partie 2.4.2). De plus, comme vu dans la partie 3.4.2 du chapitre 3, l’algorithme DNA utilisant ADMM-Reg est beaucoup plus lent à évoluer que l’algorithme DNA-APPGML. Dans ce chapitre, nous avons donc choisi d’utiliser l’algorithme de reconstruction DNA-APPGML afin de réduire les temps de calcul.

Les paramètres du DNA-APPGML sont les mêmes que ceux utilisés pour le DNA sur le fantôme cérébral du chapitre 2 (cf. partie 2.4.2.2), excepté pour le nombre d’itérations internes qui a été fixé à 2. 300 itérations externes ont été réalisées.

Intéressons nous tout d’abord à une première expérience où du bruit aléatoire ou une image anatomique est utilisée en entrée du réseau DIP, et où toutes les sauts de connexion (SC) ne sont pas forcément utilisées.

4.3 Expériences préliminaires

Dans cette partie et les suivantes, nous désignerons les méthodes utilisées avec les notations abrégées suivantes :

- MR p SC : le DNA utilise l’image IRM en entrée du réseau DIP, avec les p SC les plus intérieures. Par exemple, “MR 2 SC” est une reconstruction avec le DNA utilisant un réseau de neurones comme sur la figure 2.3 où la plus grande SC a été retirée, et avec image IRM en entrée. L’utilisation du DNA avec DIP conditionnel correspond à la méthode “MR 3 SC”,
- random p SC : de la même façon, le DNA utilise une image aléatoire tirée d’une distribution uniforme $\mathcal{U}(0, 1)$ en entrée du réseau DIP, avec les p SC les plus intérieures. Par exemple, “random 1 SC” est une reconstruction avec le DNA utilisant un réseau de neurones comme sur la figure 2.3 où les deux plus grandes SC ont été retirées, et avec image aléatoire en entrée.

Les données simulées FDG du fantôme cérébral présenté dans la partie 4.2.1 ont été reconstruites avec ces différentes méthodes. 30 itérations du DIP ont été réalisées plutôt que 200 (nombre utilisé dans le chapitre 2), au vu des résultats obtenus dans la partie 3.6.1 du chapitre 3.

L’architecture du réseau de neurones utilisé dans cette étude est présentée sur la figure 2.3 de la partie 2.3.3. Celle-ci possède une profondeur de 3, et donc 3 SC peuvent lier l’encodeur au décodeur de l’architecture.

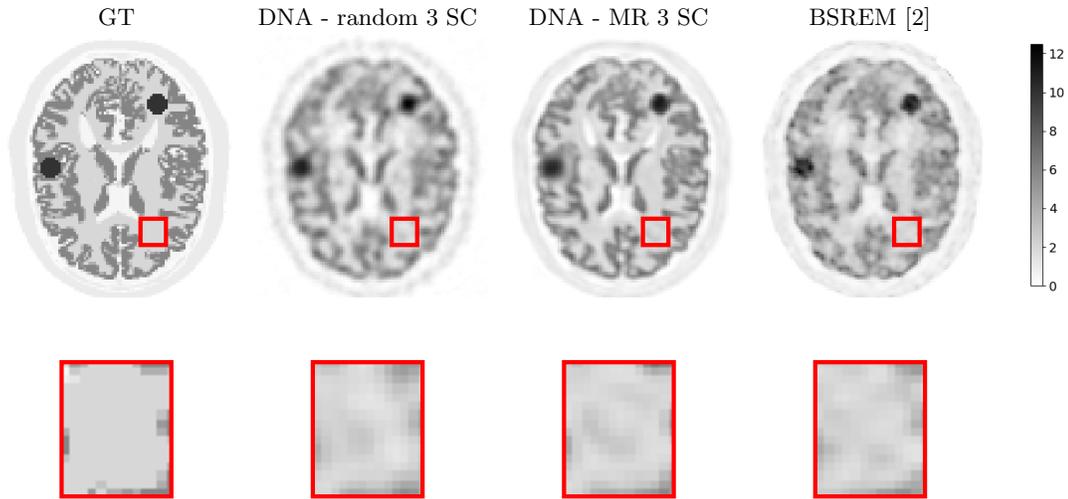


FIGURE 4.2 – Images du fantôme cérébral FDG reconstruites avec l’algorithme DNA. La première ligne montre la GT (*première image*) et différentes reconstructions : en utilisant une image aléatoire en entrée du réseau DIP et 3 SC (*deuxième image*), une image IRM en entrée du réseau DIP et 3 SC (*troisième image*), et une image proche de la routine clinique reconstruite par BSREM (*quatrième image*). La deuxième ligne montre un zoom pour ces trois images sur la zone “IRM seule”.

La figure 4.2 montre sur la première ligne les images reconstruites d’un réplicat avec les méthodes :

- “MR 3 SC” utilisée originellement par le DIPRecon [37],
- “random 3 SC” étant le paramétrage proposée dans l’article original du DIP [92],
- l’algorithme BSREM avec pénalité RDP proche de celui utilisé en clinique [2] ($\beta = 2$ a été choisi manuellement).

En regardant les images reconstruites de façon globale, l’image “MR 3 SC” présente beaucoup plus de détails et un meilleur contraste que les deux autres images de la figure 4.2. De plus, l’image produite par la méthode “random 3 SC” est plus bruitée que celle reconstruite par “MR 3 SC”, et légèrement plus que celle reconstruite par BSREM. En regardant le zoom sur la région “IRM seule” présenté sur la deuxième ligne de la figure 4.2, on voit apparaître sur l’image reconstruite par “MR 3 SC”, un artefact de forme circulaire dont les contours semblent suivre la forme de la tumeur. Ces artefacts ne sont pas visibles sur les images reconstruites avec les deux autres méthodes n’utilisant pas d’information anatomique.

La figure 4.3 montre des images reconstruites sans aucune SC, avec en entrée du DIP une image aléatoire (DNA “random 0 SC”) ou l’image anatomique (DNA “MR 0 SC”). Comparée à l’image DNA “random 3 SC” (figure 4.2 gauche), l’image DNA “random 0 SC” est moins bruitée. Comparée à l’image DNA “MR 3 SC” (figure 4.2 milieu), l’image DNA “MR 0 SC” ne présente plus d’artefact lié à la discordance entre l’image IRM et l’image TEP, comme le montre le zoom sur la zone de la tumeur “IRM seule”. Toutefois, elle comporte moins de structures anatomiques et est plus lisse. On peut déduire de ces résultats que la diminution du nombre de SC réduit l’impact de l’entrée du DIP sur la sortie.

Globalement, les résultats que l’on observe ici dans le cadre d’une reconstruction avec l’algorithme DNA sont en accord avec les résultats observés lors d’un simple débruitage DIP présentés dans le chapitre 3 (partie 3.5.6.6).

A partir de ces observations, nous avons construit deux méthodes pour mitiger les artefacts liés à la discordance entre l’image IRM utilisée en entrée du DIP et l’image TEP, tout en préservant au maximum le bénéfice qu’apporte l’image IRM à l’image TEP sur les structures communes aux deux modalités.

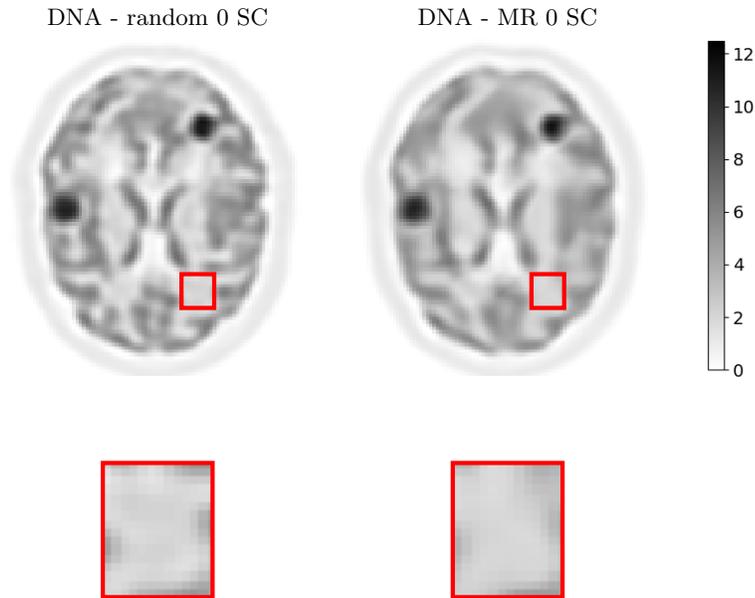


FIGURE 4.3 – Images du fantôme cérébral FDG reconstruites avec l’algorithme DNA. La première ligne montre les reconstructions en utilisant une image aléatoire en entrée du réseau DIP et 0 SC (*gauche*) et une image IRM en entrée du réseau DIP et 0 SC (*droite*). La deuxième ligne montre un zoom pour ces deux images sur la zone “IRM seule”.

4.4 Méthodes proposées pour mitiger les discordances

Nous proposons ici deux méthodes, dont le but est de diminuer les artefacts présents avec la méthode “MR 3 SC”, tout en gardant au maximum les détails visuels apportées par l’image anatomique ainsi que les propriétés quantitatives du DNA-APPGML.

Pour les deux méthodes, l’algorithme EMV a été utilisé pour l’initialisation du DNA-APPGML.

4.4.1 Méthode “MR init”

La première méthode que nous avons développée se nomme “MR init”. L’idée est de ne donner l’image anatomique en entrée du DIP que pour l’initialisation de l’algorithme de reconstruction, puis de poursuivre les débruitages sans celle-ci pour ne pas continuer à donner d’éventuelles informations discordantes à l’image TEP en cours de reconstruction (région “IRM seule” et “TEP mismatch”). Ainsi, suite à l’initialisation, lors des itérations externes de DNA, l’image anatomique en entrée du réseau DIP est remplacée par une image aléatoire avec un bruit uniforme $\mathcal{U}(0, 1)$ fixe au cours de ces itérations. Le nombre de SC du réseau DIP est fixé à 3 pour le débruitage initial avec l’image anatomique en entrée, car nous avons vu que ce paramétrage permet d’obtenir les meilleurs performances quantitatives ainsi que le plus de détails lors de l’utilisation du DIP (cf. partie 3.5.6.6). Ensuite, pour les débruitages lors de la reconstruction, le nombre de SC est fixé entre 3 et 0, en supprimant les SC de la connexion la plus longue à la connexion la plus courte.

Pour cette méthode, nous avons choisi les mêmes hyperparamètres que pour les méthodes “MR 3 SC” et “random 0 SC”, excepté 100 itérations du DIP.

4.4.2 Méthode “MR 5”

La deuxième méthode que nous proposons se nomme “MR 5”. Cette méthode est extensible à l’utilisation de Q images, mais nous avons choisi $Q = 5$ dans toute cette étude. L’idée est d’utiliser plusieurs entrées pour le réseau DIP en donnant plus ou moins d’informations IRM et d’effectuer un débruitage avec toutes les images d’entrée simultanément.

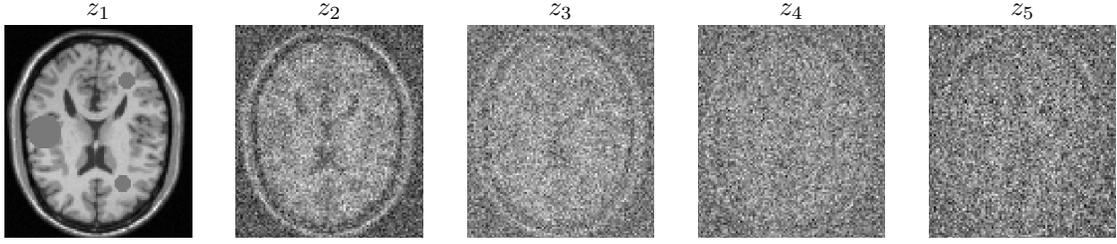


TABLE 4.1 – Les 5 images utilisées en entrée du réseau DIP avec la méthode “MR 5” et un paramètre $\Sigma = 0.1$.

La contrainte $x = f(\theta|z)$ présente dans les problèmes d’optimisation initiaux du DNA, DNA-APPGML ou du DIPRecon est remplacée par la contrainte suivante :

$$x = f(\theta|z_1, \dots, z_Q) \quad (4.1)$$

avec z_1, \dots, z_Q étant Q images utilisées en entrée du réseau DIP.

Les Q images en entrée du réseau DIP (z_q) $_{q \in [1; Q]}$ sont des versions de plus en plus bruitées de l’image anatomique mise à l’échelle (cf. partie 3.5.2) après ajout d’un bruit gaussien $\mathcal{N}(0, (q-1)\Sigma^2)$, avec Σ un paramètre pour gérer le niveau de bruit ajouté. L’image z_1 est ainsi l’image anatomique originale (sans ajout de bruit). La figure 4.1 montre ces images d’entrée pour $Q = 5$ et $\sigma = 0, 1$.

La fonction de coût pour optimiser un tel réseau DIP pour du débruitage est la même que celle présentée dans l’équation (2.2) du chapitre 2, mais en sommant sur les Q images d’entrée :

$$\theta^* = \arg \min_{\theta} \sum_{q=1}^Q \|x_{\text{label}} - f(\theta|z_q)\|_2^2, \quad x^* = f(\theta^*|z) \quad (4.2)$$

La dérivation du DNA, DNA-APPGML et DIPRecon s’étend facilement à la nouvelle contrainte, en remplaçant simplement la fonction de coût liée à l’étape de débruitage DIP à l’intérieur de ces algorithmes de reconstruction.

La nouvelle fonction de coût s’apparente à celle liée à un entraînement d’un réseau de neurones de façon supervisée, avec ainsi une “base de données” de Q paires d’images (z_q, x_{label}) $_{q \in [1; Q]}$. Néanmoins, l’entraînement du réseau DIP reste toujours non supervisé car la “base de données” ainsi construite n’utilise pas de données annotées manuellement. En effet, elle contient l’image IRM et ses versions bruitées, ainsi que l’image étiquette x_{label} à débruiter. Le réseau DIP est alors optimisé pour débruiter une image en partant de Q images différentes.

L’entraînement d’un réseau de neurones avec plusieurs paires d’images utilise des optimiseurs réalisant une descente de gradient. Ce gradient peut être approximé par le gradient d’un sous-ensemble des paires d’images, appelé batch. Ici, nous avons décidé d’optimiser le réseau DIP par mini-batch, c’est-à-dire qu’une seule paire d’images est utilisée pour chaque batch. Ainsi, pour réaliser une époque, c’est-à-dire réaliser une descente de gradient en passant une fois par chaque paire d’images du DIP lors de l’optimisation, il faut réaliser Q sous-itérations notées τ_q . A la sous-itération τ_q , le gradient de la fonction de coût (4.2) est approximé par le gradient associé à l’image d’entrée z_q .

3 SC sont utilisées pour cette méthode. Rappelons que les SC introduisent de l’information spatiale provenant de l’image d’entrée dans l’image de sortie du réseau DIP (cf. partie 4.3). En particulier, le bruit de l’image d’entrée se répercute dans l’image de sortie du réseau DIP. Ainsi, à la fin de chaque entraînement, nous avons utilisé l’image de sortie du DIP obtenue avec l’image IRM en entrée afin d’éviter d’ajouter du bruit inutilement venant des autres entrées (z_q) $_{q \in [2; Q]}$.

Les hyperparamètres choisis pour la méthode “MR 5” ($Q = 5$) sont les mêmes que pour les méthodes “MR 3 SC” et “random 0 SC”, excepté :

- le facteur de décroissance α_{EMV} a été choisi avec une valeur 5 fois plus grande que pour les méthodes utilisant une image en entrée,

- le nombre de patience P a été divisé par 5 par rapport aux méthodes utilisant une image en entrée,
- le nombre d’itérations du DIP a été diminué par rapport aux méthodes “MR 3 SC” et “random 0 SC” et a été choisi à 10.

4.5 Évaluation des méthodes proposées

4.5.1 Expériences

Les méthodes “MR init” et “MR 5” ont d’abord été chacune comparées à la méthode “MR 3 SC” qui fournit la meilleure image visuellement selon [37], et à la méthode “random 0 SC” qui comporte moins de bruit que le DIP original proposé par [92]. De plus, les méthodes “MR 3 SC” et “random 0 SC” ont données les meilleurs résultats d’un débruitage DIP en faisant varier le nombre de SC dans la partie 3.5.6.6 du chapitre 3. Ces deux méthodes seront utilisées comme référence pour étudier les performances des deux méthodes proposées.

Par la suite, les deux méthodes ont été comparées entre elles, et par rapport à la méthode BSREM avec pénalité RDP avec un paramétrage proche de celui utilisé en routine clinique [2].

Des courbes de compromis ont été réalisées comme dans les chapitres 2 et 3 dans les trois ROIs définies dans la partie 4.2.1, en fonction des itérations externes des différents algorithmes. La récupération d’activité a été calculée dans les deux tumeurs TEP. Le biais relatif a été calculé dans la LCR et la région “IRM seule”, correspondant à de la matière blanche dans l’image TEP. Le SSIM a été calculé dans l’ensemble du cerveau. La rugosité de l’image a été calculée dans la matière blanche du cerveau mais à laquelle nous avons retiré les voxels de la région “IRM seule”. Une inspection visuelle a été effectuée pour l’évaluation qualitative des méthodes. Enfin, nous avons analysé des profils du biais relatif à la GT TEP à travers les tumeurs.

Nous avons analysé des profils du biais relatif aux valeurs TEP au centre de chaque tumeur. Pour la ROI “IRM seule”, la valeur de la GT est celle de la matière blanche le long de toute la ligne (ici 2). Pour les régions “TEP match” et “TEP mismatch”, la GT vaut 10 à l’intérieur de la tumeur TEP, mais est plus faible sur les extrémités de la ligne considérée.

Voyons dans la suite les résultats obtenus, d’abord avec la méthode “MR init”, puis la méthode “MR 5”, et enfin la comparaison entre les deux.

4.5.2 Résultats

4.5.2.1 Résultats avec la méthode “MR init”

La figure 4.4 montre les images reconstruites pour un réplicat avec les deux méthodes de référence (“MR 3 SC” et “random 0 SC”), et la méthode “MR init” utilisé avec 0 SC ou 2 SC. Les images de cette figure ne sont pas choisies après 300 itérations externes, mais plutôt pour avoir des rugosités de l’image dans la matière blanche proches, ainsi que le maximum de correspondance visuelle avec la GT. Les images obtenues avec la méthode “MR init” sont assez similaires avec celle obtenue avec la méthode “random 0 SC”, et présente beaucoup moins de détails dans le cortex par rapport à la méthode “MR 3 SC”. En revanche, en regardant les zooms sur la région “IRM seule”, seule la méthode “MR 3 SC” présente un artefact dans cette ROI (la méthode “MR init” l’ayant supprimé).

La figure 4.5 montre la récupération d’activité dans les zones d’intérêt des tumeur “TEP match” et “TEP mismatch”, le biais relatif dans la ROI “IRM seule” et la zone d’intérêt du LCR, en fonction de la rugosité de l’image dans la matière blanche pour toutes les méthodes comparées. Dans la région “IRM seule” (en bas à gauche de la figure 4.5), la méthode “MR 3 SC” se caractérise par une faible rugosité d’image mais un certain biais, tandis que la méthode “random 0 SC” a un biais presque nul mais une rugosité d’image plus élevée. Quant aux courbes produites avec la méthode “MR init”, quel que soit le nombre de SC, celles-ci se situent entre les courbes des deux méthodes de référence. Pour les dernières itérations de la méthode “MR

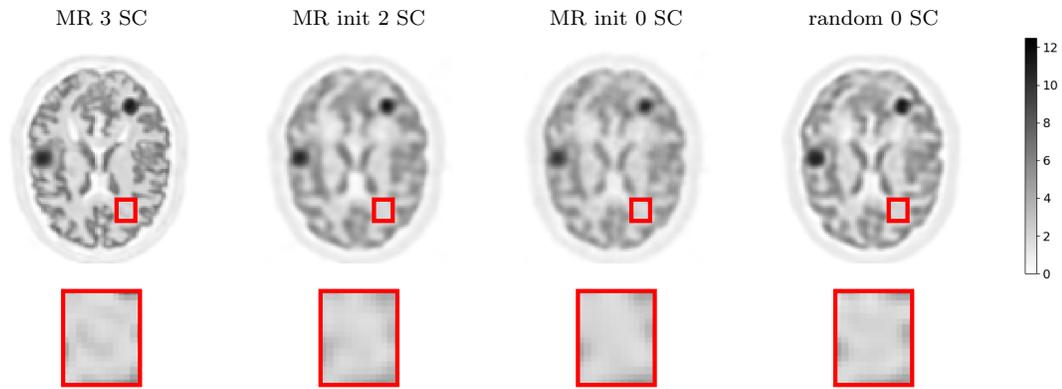


FIGURE 4.4 – Images reconstruites par le DNA-APPGML en utilisant la méthode “MR init” pour différents nombre de SC (*deuxième et troisième images*), en comparaison avec le DIP conditionnel (“MR 3 SC” *gauche*) et une entrée aléatoire du réseau DIP et 0 SC (“random 0 SC” *droite*).

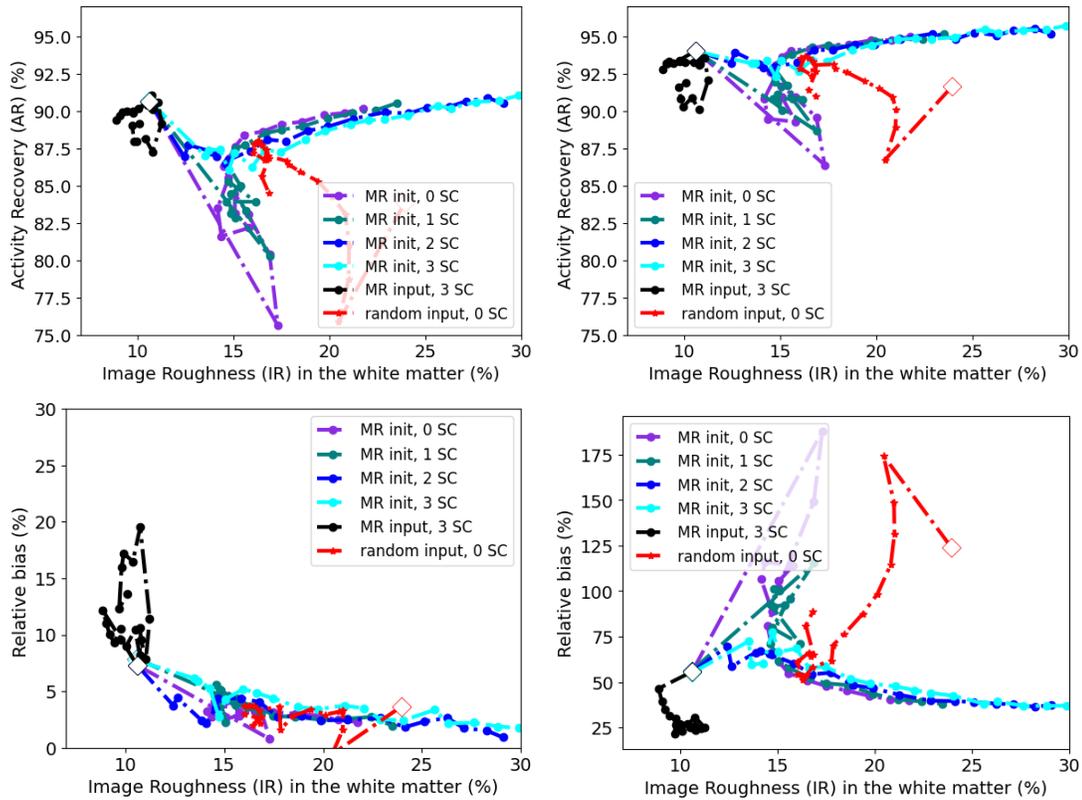


FIGURE 4.5 – Courbes de compromis moyennées sur 40 réplicats de la récupération d’activité dans la ROI tumeur match (*haut gauche*) et la ROI tumeur mismatch (*haut droite*), du biais relatif dans la ROI “IRM seule” (*bas gauche*) et du biais relatif dans la ROI LCR (*bas droite*), en fonction de la rugosité dans l’ensemble de la matière blanche, pour l’algorithme DNA-APPGML proposé. La méthode “MR init” a été utilisée avec 0, 1, 2 ou 3 SC, et a été comparée à l’image IRM en entrée du DIP et 3 SC ainsi qu’à une entrée aléatoire avec 0 SC. Les diamants blancs correspondent à l’initialisation du DNA-APPGML.

init”, la rugosité de l’image continue d’augmenter. De plus, pour la méthode “MR init”, plus le nombre de SC est élevé, plus la rugosité de l’image augmente rapidement avec les itérations.

Dans la région chaude “TEP match” (en haut à gauche de la figure 4.5), la méthode “MR 3 SC” obtient la meilleure récupération d’activité pour la meilleure rugosité de l’image à l’initialisation, puis n’évolue pas beaucoup. La méthode “random 0 SC” est caractérisé par une initialisation avec une rugosité de l’image élevée par rapport aux autres méthodes. Celle-ci diminue au cours des itérations en atteignant une récupération d’activité légèrement inférieure à la méthode “MR 3 SC”. La rugosité de l’image pour la méthode “MR init” augmente au fur et à mesure des itérations, avec une récupération d’activité qui diminue légèrement puis augmente de nouveau avec la rugosité de l’image. On remarque des régimes plus chaotiques pendant les premières itérations pour la méthode “MR init” avec 0 ou 1 SC.

Pour la région “TEP mismatch” (en haut à droite de la figure 4.5), les observations sont les mêmes que dans la région “TEP match” mais avec moins de différences entre les méthodes. Enfin, le biais relatif dans la région froide (en bas à droite de la figure 4.5) diminue le plus pour la méthode “MR 3 SC” par rapport à l’initialisation et diminue légèrement pour la méthode “MR init” quelque soit le nombre de SC, en atteignant une valeur toujours plus faible que pour la méthode “random 0 SC”.

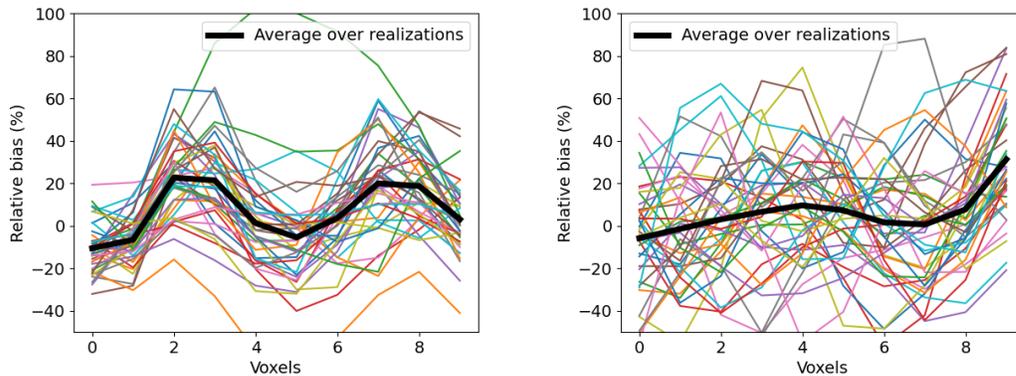


FIGURE 4.6 – Profils le long d’une ligne horizontale au centre de la ROI “IRM seule” montrant le biais relatif de chaque voxel de l’image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affichée dans une couleur. Méthode “MR 3 SC” (*gauche*) et méthode “MR init 2 SC” (*droite*).

La figure 4.6 montre un profil du biais relatif à la GT le long d’une ligne dans la tumeur “IRM seule” pour la méthode “MR 3 SC” (*gauche*) et “MR init” avec 2 SC (*droite*). Pour la méthode “MR 3 SC”, on observe deux pics d’intensité biaisés positivement, contrairement à la méthode “MR init” où on observe une ligne relativement plate et non biaisée.

De même, la figure 4.7 montre un profil du biais relatif à la GT le long d’une ligne dans la tumeur “TEP mismatch” pour la méthode “MR 3 SC” (*gauche*) et “MR init” avec 2 SC (*droite*). Pour les deux graphes, on remarque un comportement similaire, avec les voxels au centre de la tumeur TEP qui sont non biaisés. Les voxels situés aux extrémités des deux lignes sont biaisés négativement, avec un changement de valeur qui semble légèrement plus marqué pour la méthode “MR init” reflétant des contours légèrement moins lissés.

4.5.2.2 Résultats avec la méthode “MR 5”

NB : Dans cette partie, les résultats sont présentées de la même façon que les résultats de la méthode “MR init”.

La figure 4.8 montre les images reconstruites pour un réplicat avec les deux méthodes de référence et la méthode “MR 5”. Les images obtenues avec la méthode “MR 5” présente visuellement moins de détails que la méthode “MR 3 SC”, mais plus de détails que la méthode

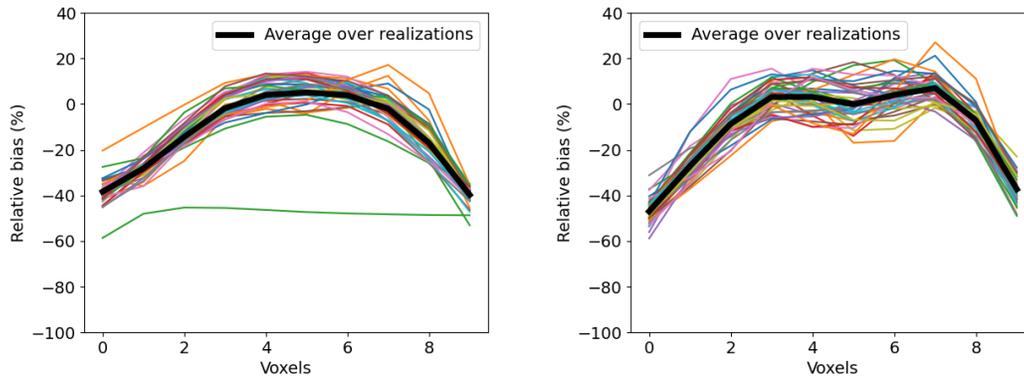


FIGURE 4.7 – Profil le long d’une ligne horizontale au centre de la ROI “TEP mismatch” montrant le biais relatif de chaque voxel de l’image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode “MR 3 SC” (*gauche*) et méthode “MR init 2 SC” (*droite*).

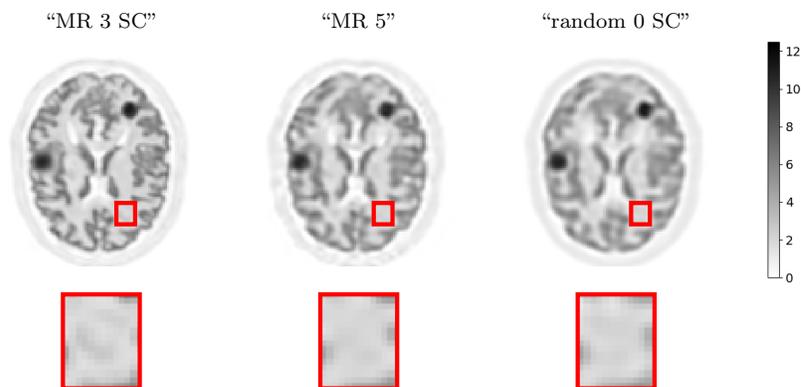


FIGURE 4.8 – Images reconstruites par le DNA-APPGML en utilisant la méthode “MR 5” (*milieu*), en comparaison avec le DIP conditionnel (“MR 3 SC” *gauche*) et une entrée aléatoire du réseau DIP et 0 SC (“random 0 SC” *droite*).

“random 0 SC”. De plus, aucun artefact n’est décelable en zoomant sur la ROI “IRM seule” contrairement à la méthode “MR 3 SC”.

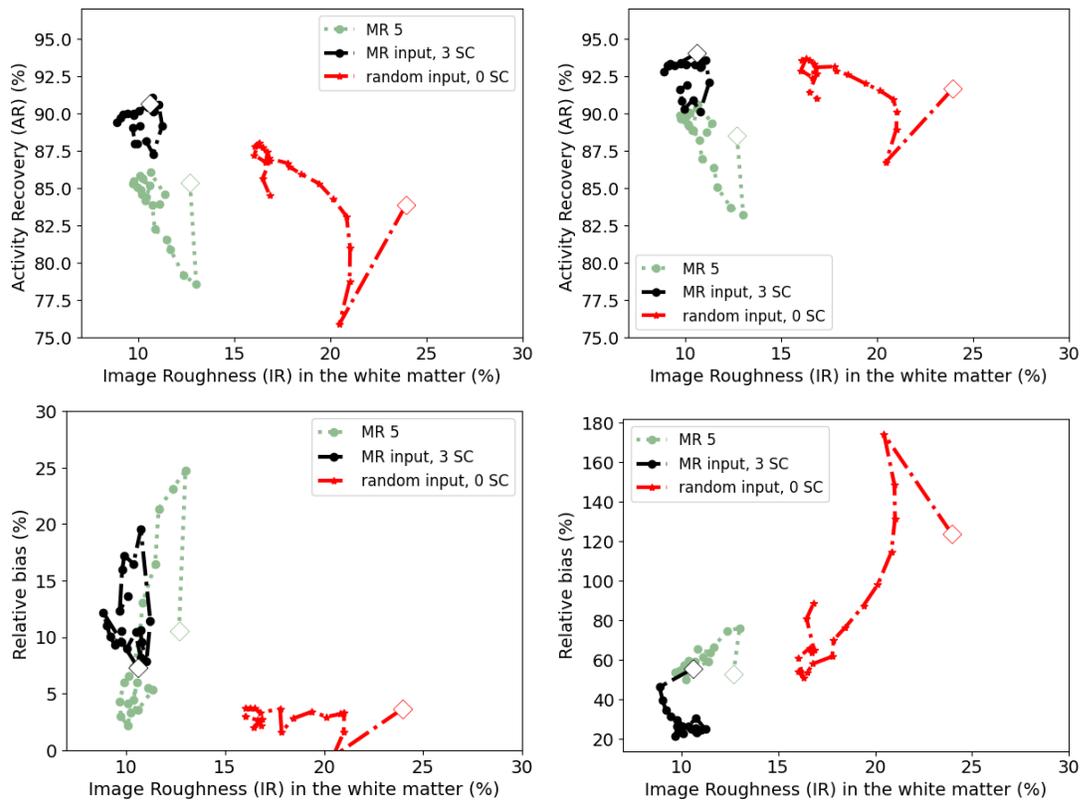


FIGURE 4.9 – Courbes de compromis moyennées sur 40 réplicats de la récupération d’activité dans la ROI tumeur match (*haut gauche*) et la ROI tumeur mismatch (*haut droite*), du biais relatif dans la ROI “IRM seule” (*bas gauche*) et du biais relatif dans la ROI LCR (*bas droite*), en fonction de la rugosité dans l’ensemble de la matière blanche, pour l’algorithme DNA-APPGML proposé. La méthode “MR 5” a été comparée à l’image IRM en entrée du DIP et 3 SC ainsi qu’à une entrée aléatoire avec 0 SC. Les diamants blancs correspondent à l’initialisation du DNA-APPGML.

La figure 4.9 montre la récupération d’activité dans les zones d’intérêt des tumeurs “TEP match” et “TEP mismatch”, le biais relatif dans la ROI “IRM seule” et la zone d’intérêt du LCR, en fonction de la rugosité de l’image dans la matière blanche pour toutes les méthodes comparées. Dans la région “IRM seule” (en bas à gauche de la figure 4.9), la méthode “MR 5” est caractérisée par une rugosité de l’image faible dès l’initialisation. Le biais relatif est du même niveau que celui de la méthode “MR 3 SC” à l’initialisation, puis augmente et diminue jusqu’à un niveau comparable à celui de la méthode “random 0 SC”, pour une rugosité de l’image plus faible.

Dans la région chaude “TEP match” (en haut à gauche de la figure 4.9), pour la méthode “MR 5”, la récupération d’activité diminue rapidement puis augmente de nouveau jusqu’à atteindre le niveau de l’initialisation mais pour une rugosité de l’image encore plus faible. La récupération d’activité atteinte est similaire à celle atteinte par la méthode “random 0 SC” et plus faible que celle atteinte par la méthode “MR 3 SC”. Le même comportement est observé dans la région “TEP mismatch” (en haut à droite de la figure 4.9), sauf que la récupération d’activité atteinte par la méthode “MR 5” est plus faible que celle atteinte par la méthode “random 0 SC”. Enfin, le biais relatif dans la région froide (en bas à droite de la figure 4.9) pour la méthode “MR 5” augmente après l’initialisation, puis diminue et se stabilise autour de la même valeur de biais qu’à l’initialisation. La valeur de biais atteinte est similaire à celle atteinte par la méthode

“random 0 SC” et supérieure à celle atteinte par la méthode “MR 3 SC”.

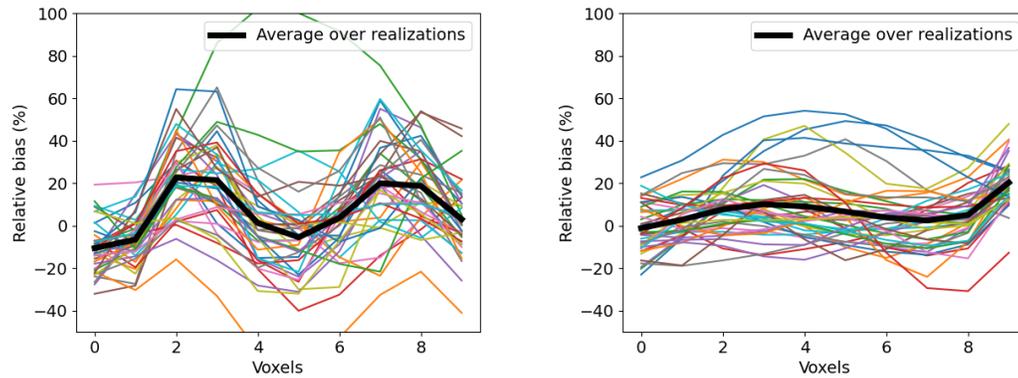


FIGURE 4.10 – Profil le long d’une ligne horizontale au centre de la ROI “IRM seule” montrant le biais relatif de chaque voxel de l’image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode “MR 3 SC” (*gauche*) et méthode “MR 5” (*droite*).

La figure 4.10 montre un profil du biais relatif à la GT le long d’une ligne dans la tumeur “IRM seule” pour la méthode “MR 3 SC” (*gauche*) et “MR 5” (*droite*). Tout comme pour la méthode “MR init” (figure 4.6), la méthode “MR 5” présente un profil relativement plat et quasiment non biaisé contrairement à la méthode “MR 3 SC” qui exhibe deux pics.

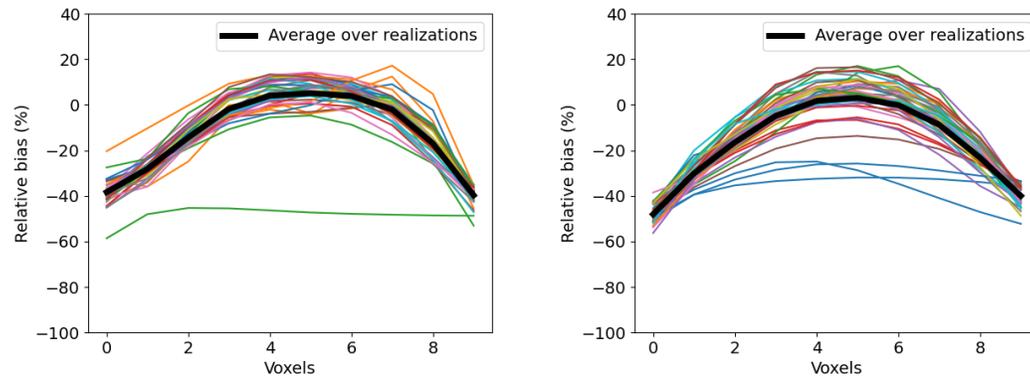


FIGURE 4.11 – Profil le long d’une ligne horizontale au centre de la ROI “TEP mismatch” montrant le biais relatif de chaque voxel de l’image reconstruite par rapport à la GT. La moyenne des profils sur 40 réplicats est affichée en noir épais, chaque réplicat est affiché dans une couleur. Méthode “MR 3 SC” (*gauche*) et méthode “MR 5” (*droite*).

De même, la figure 4.11 montre un profil du biais relatif à la GT le long d’une ligne dans la tumeur “TEP mismatch” pour la méthode “MR 3 SC” (*gauche*) et “MR 5” (*droite*). On ne note pas de différence claire entre celles-ci.

4.5.2.3 Comparaison des deux méthodes proposées

La figure 4.12 montre la récupération d’activité dans les zones d’intérêt des tumeur “TEP match” et “TEP mismatch”, le biais relatif dans la ROI “IRM seule” et la zone d’intérêt du LCR, en fonction de la rugosité de l’image dans la matière blanche. Les deux méthodes proposées

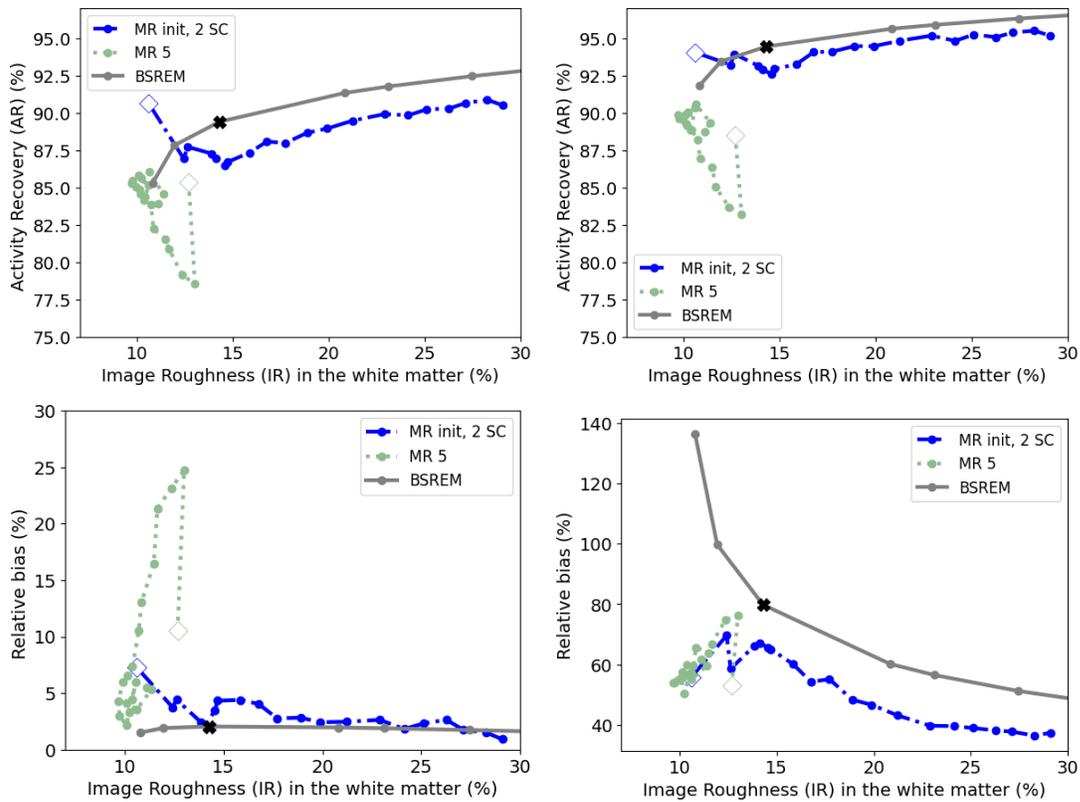


FIGURE 4.12 – Courbes de compromis moyennées sur 40 réplicats de la récupération d’activité dans la ROI tumeur match (*haut gauche*) et la ROI tumeur mismatch (*haut droite*), du biais relatif dans la ROI “IRM seule” (*bas gauche*) et du biais relatif dans la ROI LCR (*bas droite*), en fonction de la rugosité dans l’ensemble de la matière blanche, pour l’algorithme DNA-APPGML. La méthode “MR init” a été utilisée avec 2 SC et est comparée à la méthode “MR 5” et à BSREM+RDP avec différentes valeurs de pénalité. Le BSREM de départ utilisé pour DNA-APPGML correspond à la croix noire. Les diamants blancs correspondent à l’initialisation du DNA-APPGML.

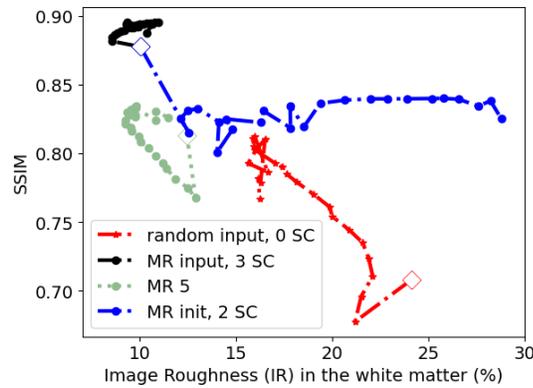


FIGURE 4.13 – SSIM dans l’ensemble du cerveau en fonction de la rugosité de l’image dans l’ensemble de la matière blanche, moyenné sur 40 réplicats, pour l’algorithme DNA-APPGML proposé. La méthode “MR init” a été utilisée avec 2 SC et est comparée à la méthode “MR 5”, ainsi qu’aux méthodes avec image IRM en entrée et 3 SC (“MR 3 SC”) et une entrée aléatoire avec 0 SC (“random 0 SC”). Les diamants blancs correspondent à l’initialisation du DNA-APPGML.

sont comparées à l’algorithme BSREM avec pénalité RDP pour différentes valeurs de pénalité. La méthode “MR init” présentée ici est celle avec 2 SC.

Dans la région “IRM seule” (en bas à gauche de la figure 4.12), on observe que les deux méthodes proposées obtiennent les mêmes valeurs de biais relatif que l’algorithme BSREM, proches de zéro. Dans les deux régions tumorales visibles en TEP (en haut de la figure 4.12), l’initialisation de la méthode “MR init” obtient la même récupération d’activité que l’image BSREM de départ mais avec une rugosité de l’image plus faible. L’initialisation de la méthode “MR 5” est quant à elle moins bonne, avec une rugosité de l’image légèrement plus élevée et une récupération d’activité plus faible. Pendant la reconstruction, la méthode “MR init” a une récupération d’activité plus faible que la méthode BSREM pour les mêmes valeurs de rugosité de l’image. La méthode “MR 5” quant à elle diminue la rugosité de l’image jusqu’à stagner, pour une récupération d’activité quasiment identique avec BSREM, mais moins élevée que pour l’initialisation de la méthode “MR init”. Enfin, dans la région froide du LCR (en bas à droite de la figure 4.12), les deux méthodes proposées ont un biais relatif comparable pour un même niveau de rugosité de l’image, et inférieur à BSREM. De plus, la méthode “MR init” arrive à faire encore diminuer le biais relatif avec l’augmentation de la rugosité de l’image, qui est meilleur que celui de l’algorithme BSREM.

La figure 4.13 montre le SSIM dans l’ensemble du cerveau en fonction de la rugosité de l’image dans la matière blanche pour les méthodes proposées ainsi que les méthodes de référence. Le SSIM de la méthode “MR 5” diminue rapidement puis augmente de nouveau pour se stabiliser au même niveau que le SSIM de la méthode “MR init”. La méthode “MR 3 SC” présente toujours quant à elle un SSIM plus élevé que les deux méthodes proposées. Inversement, la méthode “random 0 SC” a un SSIM qui augmente lorsque la rugosité de l’image diminue, mais arrive à un palier puis diminue. Le palier est toujours plus bas que le meilleur SSIM atteint par les deux méthodes proposées.

4.6 Discussion

La méthode “MR 3 SC” est celle reconstruisant une image visuelle la plus intéressante, avec des détails venant de l’image anatomique utilisée en entrée du réseau DIP. Elle obtient aussi le meilleur SSIM pour un niveau de rugosité de l’image faible, parmi toutes les méthodes étudiées. Ces observations rejoignent les résultats obtenus par les méthodes utilisant du DIP conditionnel [37] plutôt qu’une image aléatoire en entrée du réseau DIP. Cependant, dans la région “IRM seule”, les bords de la tumeur sont présents dans l’image reconstruite par la méthode “MR 3

SC” et constitue un artefact non désirable en pratique clinique. Celui-ci a été mis en évidence de façon quantitative par le biais relatif positif dans cette région, ainsi que par deux pics d’intensité plus élevés au niveau des bords de la tumeur sur la figure de profil dans cette région (graphe de gauche dans la figure 4.6).

En revanche, comme l’ont montré les images reconstruites, ainsi que les courbes de biais relatif et les profils dans la région “IRM seule”, la méthode “MR 5” parvient à supprimer l’artefact. C’est aussi le cas de la méthode “MR init” peu importe le nombre de SC.

Pour la méthode “MR init”, le nombre de SC a un impact sur la reconstruction. D’une part, avec 0 ou 1 SC, nous avons observé que le comportement de l’algorithme de reconstruction est instable dans les premières itérations. D’autre part, pour la méthode “MR init” avec 3 SC, le bruit s’ajoute plus rapidement au fil des itérations, comme le montre la rugosité de l’image. Cela est dû à l’utilisation de toutes les SC combinées à une entrée aléatoire lors de la reconstruction. Le bruit de l’entrée est alors injecté dans l’image de sortie du réseau DIP. Par la suite, la méthode “MR init” avec 2 SC a été comparée à la méthode “MR 5” pour ces raisons.

Les deux méthodes proposées permettent d’éliminer efficacement l’artefact observé dans la région “IRM seule” avec la méthode “MR 3 SC”, tout en améliorant les métriques obtenues par l’autre méthode de référence “random 0 SC”. En effet, les deux méthodes ont permis d’obtenir des images avec une rugosité de l’image plus faible qu’avec la méthode “random 0 SC”. D’une part, la méthode “MR 5” permet clairement d’avoir une image visuellement de meilleure qualité que celle produite par la méthode “random 0 SC”. D’autre part, la méthode “MR init” permet d’obtenir de meilleurs métriques dans les régions chaudes et la région froide du fantôme que la méthode “random 0 SC”.

Finalement, les deux méthodes proposées résolvent le problème de l’artefact dans la région “IRM seule”, tout en ayant des propriétés différentes. D’une part, la méthode “MR init” permet d’obtenir dans tous les cas une meilleure récupération d’activité que la méthode “MR 5” dans les deux régions tumorales. Elle peut même l’améliorer en augmentant la rugosité de l’image, contrairement à la méthode “MR 5” qui obtient une valeur de récupération d’activité assez proche à chaque itération. D’autre part, la méthode “MR 5” produit des images visuellement plus proches de la méthode “MR 3 SC”, permettant de reconstruire des images avec peu de bruit et sans l’artefact dans la région “IRM seule”.

De plus, les profils dans la région “TEP mismatch” sont non biaisés au centre de la tumeur pour la méthode “MR 3 SC” ainsi que les deux méthodes proposées. Un changement d’allure est attendu au niveau des voxels en dehors de la tumeur TEP car le biais relatif est calculé par rapport à la valeur dans la tumeur TEP. La méthode “MR init” semble exhiber des bords plus définis que les méthodes “MR 3 SC” et “MR 5” d’après les figures de profil, car l’information anatomique n’est présente qu’à l’initialisation de l’algorithme de reconstruction. La tumeur étant plus grande dans l’image anatomique, cela impacte plus la netteté des bords dans l’image TEP pour les méthodes “MR 3 SC” et “MR 5”.

4.7 Conclusion

L’utilisation d’une image anatomique en entrée d’un réseau DIP convolutionnel avec toutes les SC permet d’obtenir une image visuellement attrayante.

Cette méthode a été utilisée aussi bien en débruitage qu’au sein de la reconstruction TEP car les SC permettent d’apporter des détails venant de l’information anatomique vers l’image TEP. Toutefois, des artefacts potentiellement gênants en pratique clinique peuvent apparaître dans le cas de tumeurs présente uniquement dans l’image anatomique. Pour y remédier, nous avons proposé deux méthodes qui éliminent efficacement les artefacts dans le cas de ce type de tumeur. Cependant, un compromis existe entre le niveau de détails dans l’image reconstruite, la quantification des régions chaudes et froides et le niveau de bruit. La méthode “MR init” a un comportement stable ainsi qu’un niveau de bruit raisonnable pour 2 SC, et permet une bonne quantification. Quant à la méthode “MR 5”, elle produit visuellement une meilleure image et permet de supprimer l’artefact pour un niveau de bruit faible que la méthode “MR init”. Le choix de la méthode dépend de l’application et du compromis désiré.

Discussion générale - Conclusion - Perspectives

Dans le chapitre 2, nous avons développé l'algorithme de reconstruction DNA comme étant d'abord une combinaison du DIPRecon et d'ADMM-Reg, deux algorithmes de reconstruction TEP récemment proposés dans la littérature. Le premier a comme avantage d'utiliser le concept récent du DIP pour pallier la nature mal posée du problème de reconstruction TEP tout en permettant d'introduire de l'information anatomique généralement riche en détails structurels. Le second propose une dérivation mathématique élégante pour autoriser des valeurs négatives dans l'image reconstruite, permettant de réduire le biais caractéristique des méthodes avec contrainte de non-négativité, sans affecter l'hypothèse sur la nature Poissonienne des données TEP. L'algorithme DNA a donc été développé dans le but de tirer le meilleur parti de ces deux algorithmes. Ses performances ont été comparées à celles de DIPRecon, d'ADMM-Reg et d'autres algorithmes de l'état de l'art. Les résultats obtenus sur des simulations 2D FDG de cerveau et d'un fantôme cylindrique d' ^{90}Y , ainsi que sur des données réelles 3D FDG de cerveau, ont montré que le DNA peut reconstruire des images de haute qualité, avec un niveau de bruit très faible en FDG, et acceptable en ^{90}Y , en comparaison avec les autres algorithmes. Ces images présentent un niveau de détails élevé grâce à l'apport d'information anatomique par le DIP qui est plus naturellement incorporée dans la reconstruction TEP en comparaison avec d'autres méthodes comme celles utilisant une pénalité MRF avec des poids de Bowsher. Les résultats quantitatifs sont de très bon niveau dans les régions tumorales par rapport aux autres algorithmes de reconstruction étudiés, en particulier grâce à la très bonne initialisation de l'algorithme que nous avons proposée. Plus précisément, celle-ci part d'une image reconstruite par l'algorithme BSREM avec pénalité RDP qui est légèrement bruitée mais qui présente un bon contraste. Cette image BSREM de départ est débruitée de façon convaincante grâce au DIP qui est arrêté automatiquement avant convergence par l'algorithme EMV. Quant aux résultats dans les régions froides, le DNA réalise une réduction de biais par rapport aux algorithmes ayant une contrainte de non-négativité dans l'image, comparable à celle obtenue ADMM-Reg. Au delà de cette association efficace des deux algorithmes, ce travail a permis de mettre en lumière les artefacts causés par la fonction d'activation finale ReLU de l'algorithme DIPRecon, ainsi que des imprécisions théoriques sur le problème d'optimisation résolu par ADMM-Reg (cf. Annexe B.3).

Le DNA comprend plusieurs hyperparamètres impactant son comportement ainsi que la qualité des images que le réseau DIP peut produire. Chaque hyperparamètre a été étudié à partir d'expériences de débruitage DIP ou de reconstruction avec le DNA au sein du chapitre 3. En comprenant mieux leur rôle, certains hyperparamètres ont pu être automatisés. C'est le cas du paramètre de pénalité ADMM ρ_2 qui est spécifique au DNA et supplémentaire par rapport au DIPRecon. Il n'est donc pas une faiblesse pour le DNA. Par ailleurs, l'utilisation du DIP dans les algorithmes DIPRecon et DNA rend le problème d'optimisation non convexe. L'initialisation a alors beaucoup d'importance sur les performances de ces algorithmes. Dans ce travail, nous avons automatisé le processus d'initialisation de ces algorithmes, qui plus est se révèle proche de l'initialisation optimale. L'étude des hyperparamètres a aussi mis en évidence l'interdépendance de plusieurs paramètres. C'est le cas notamment du paramètre de pénalité ADMM ρ_1 et du nombre d'itérations internes, ainsi que le nombre d'itérations du DIP dans la reconstruction. Nous avons pu donner des recommandations sur la manière de régler ces hyper-

paramètres de façon conjointe et éviter ainsi des situations sous-optimales, voire de divergence. En outre, l'étude des hyperparamètres spécifiques au DIP a permis de mieux comprendre les choix conduisant à de bonnes propriétés du DIP au sein de la reconstruction. Nous avons pu obtenir des conclusions claires grâce aux expériences : l'optimiseur Adam est le plus adapté, avec un taux d'apprentissage de 0,01, la méthode d'initialisation des poids n'a pas d'importance et la méthode de mise à l'échelle est fonction de l'algorithme de reconstruction. Ces expériences ont montré que certains choix de la littérature justifiés par intuition ne sont pas optimaux, comme par exemple l'utilisation de l'optimiseur L-BFGS dans [37]. Enfin, l'étude détaillée des hyperparamètres des algorithmes utilisés dans la littérature n'est pas toujours réalisée. Pourtant, celle-ci s'avère cruciale pour obtenir les meilleures performances d'un algorithme. Dans ce travail, les conclusions de l'étude des hyperparamètres dans le chapitre 3 n'ont pas toujours été appliquées aux expériences des chapitres 2 et 4. Les résultats présentés dans ces deux chapitres pourraient donc être davantage optimisés.

Nous avons étudié précisément l'impact du choix de l'architecture du réseau DIP et de son entrée, dans le cadre d'un débruitage simple ainsi que dans le cadre de la reconstruction. Premièrement, dans le cas où le réseau DIP est utilisé sans information anatomique en entrée, une architecture de type U-net convolutionnelle (telle qu'utilisée par le DNA dans le chapitre 2) sans sauts de connexion (SC), est la plus adaptée pour un débruitage DIP. Dans nos expériences, l'ajout à chaque itération d'un bruit aléatoire modéré dans l'image d'entrée a permis d'améliorer légèrement les capacités de débruitage, ce qui n'est pas le cas de l'ajout de couches de contrôle spectral. Deuxièmement, dans le cas où le réseau DIP utilise une image anatomique en entrée et où il y a concordance entre information anatomique et image TEP, la même architecture de type U-net convolutionnelle mais avec des SC à toutes les profondeurs (comme utilisée dans les chapitres 2 et 4) a donné les meilleures performances quantitatives et qualitatives dans l'étude sur le débruitage DIP. L'utilisation de transformateurs, proposés plus récemment dans le domaine de l'intelligence artificielle, n'a pas permis d'améliorer les capacités de débruitage. Dans le cas de discordances entre l'information anatomique et la TEP (notamment dans le cas d'une tumeur apparaissant uniquement sur l'image IRM), nous avons vu que l'architecture utilisée avec 3 SC dans le DIP engendre des artefacts sur l'image TEP reconstruite. Nous avons alors développé deux méthodes, "MR init" et "MR 5", toutes deux capables de supprimer les artefacts venant d'une tumeur uniquement visible dans l'image anatomique. Ces méthodes ont montré de meilleures performances que l'architecture sans SC utilisée avec une image d'entrée aléatoire. Un compromis est à faire pour choisir l'une ou l'autre méthode. "MR init" permet d'obtenir une meilleure quantification globale que "MR 5", mais "MR 5" permet de conserver davantage de détails anatomiques avec le plus faible bruit.

Le DIPRecon et le DNA possèdent des hyperparamètres en commun. Les expériences du chapitre 3 et 4 devraient être réalisées également avec l'algorithme DIPRecon pour confirmer les choix réalisés. En particulier, pour une utilisation en routine clinique des algorithmes DIPRecon et DNA, une étude plus en profondeur du nombre d'itérations externes notamment est nécessaire. En effet, ces algorithmes doivent être arrêtés avant convergence pour ne pas introduire trop de bruit dans les images reconstruites. De manière plus générale, il n'existe pas de règle à ce jour pour arrêter avant convergence les algorithmes ML. Quant aux algorithmes PML, le choix du paramètre de pénalité reste une difficulté. Dans nos expériences, nous avons vu que l'algorithme EMV a permis de trouver automatiquement une image optimale dans le cas simple du débruitage avec le DIP. En outre, le DNA (où ADMM-Reg est utilisé pour limiter le biais intrinsèque du DIP avec ReLU) pourrait théoriquement être dérivé comme un algorithme DIP de bout en bout [47]. Cela permettrait de retirer certains hyperparamètres liés à l'algorithme ADMM (nombre d'itérations internes, paramètre de pénalité ρ_2). De plus, l'EMV serait alors un candidat pour choisir de manière automatique le nombre d'itérations du DIP et ainsi déterminer de manière automatique l'image finale reconstruite.

En outre, une méthode reconstruisant une image avec des artefacts ne serait pas utilisable en clinique. Les méthodes "MR init" et "MR 5" développées dans le chapitre 4 sont plus performantes que la méthode random 0 SC. Elles permettent d'utiliser une information anatomique

sans apporter d'artefact contrairement au DIP conditionnel utilisé dans [37] dans le cas de discordances de contour entre image anatomique et image TEP. L'ajout de bruit aléatoire à l'entrée de la méthode "MR init" à chaque itération externe pourrait être étudié pour essayer de diminuer le bruit dans l'image de sortie. Une nouvelle méthode cherchant à combiner les méthodes "MR 5" et "MR init" pourrait aussi être proposée, par exemple en ajoutant de plus en plus de bruit à l'image d'entrée anatomique à chaque itération externe du DNA. Ces méthodes, tout comme le DNA, peuvent être réécrites comme des méthodes de reconstruction DIP de bout en bout.

En définitive, le transfert des derniers algorithmes proposés à des méthodes de bout en bout rendrait leur utilisation en clinique plus accessible avec un critère d'arrêt automatique pour reconstruire une image. Ces méthodes ont l'avantage de pouvoir améliorer l'image TEP grâce à une image anatomique, tout en restant fidèle à la réalité TEP, ce qui est primordial en routine clinique.

Bibliographie

- [1] S. Ahn and J. A. Fessler, “Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms,” *IEEE transactions on medical imaging*, vol. 22, no. 5, pp. 613–626, 2003.
- [2] S. Ahn, S. G. Ross, E. Asma, J. Miao, X. Jin, L. Cheng, S. D. Wollenweber, and R. M. Manjeshwar, “Quantitative comparison of osem and penalized likelihood image reconstruction using relative difference penalties for clinical pet,” *Physics in Medicine & Biology*, vol. 60, no. 15, p. 5733, 2015.
- [3] H. Alva-Sánchez, A. Zepeda-Barrios, V. Díaz-Martínez, T. Murrieta-Rodríguez, A. Martínez-Dávalos, and M. Rodríguez-Villafuerte, “Understanding the intrinsic radioactivity energy spectrum from 176lu in lyso/lso scintillation crystals,” *Scientific reports*, vol. 8, no. 1, p. 17310, 2018.
- [4] R. Badawi and P. Marsden, “Developments in component-based normalization for 3d pet,” *Physics in medicine & biology*, vol. 44, no. 2, p. 571, 1999.
- [5] B. Bai, Q. Li, C. Holdsworth, E. Asma, Y.-C. Tai, A. Chatziioannou, and R. M. Leahy, “Model-based normalization for iterative 3d pet image reconstruction,” *Physics in medicine & biology*, vol. 47, no. 15, p. 2773, 2002.
- [6] H. H. Barrett and K. J. Myers, *Foundations of image science*. John Wiley & Sons, 2013.
- [7] M. Belzunce, A. Mehranian, Z. Chalampalakakis, and A. Reader, “Evaluation of shift-invariant image-space psfs for the biograph mmr pet scanner,” in *PSMR Conference*, 2017.
- [8] C. Betancourt, A. Dätwyler, P. Owen, A. Puig, and N. Serra, “Sipm single photon time resolution measured via bi-luminescence,” *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 958, p. 162851, 2020.
- [9] S. Borman, “The expectation maximization algorithm—a short tutorial,” *Submitted for publication*, vol. 41, 2004.
- [10] A. Bousse, M. Courdurier, É. Émond, K. Thielemans, B. F. Hutton, P. Irarrazaval, and D. Visvikis, “Pet reconstruction with non-negativity constraint in projection space : Optimization through hypo-convergence,” *IEEE transactions on medical imaging*, vol. 39, no. 1, pp. 75–86, 2019.
- [11] A. K. Bousselham, “Fpga based data acquisition and digital pulse processing for pet and spect,” Ph.D. dissertation, Fysikum, 2007.
- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [13] C. Byrne, “Iterative algorithms for deblurring and deconvolution with constraints,” *Inverse Problems*, vol. 14, no. 6, p. 1455, 1998.

- [14] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet : Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*. Springer, 2022, pp. 205–218.
- [15] T. Carlier, T. Eugène, C. Bodet-Milin, E. Garin, C. Ansquer, C. Rousseau, L. Ferrer, J. Barbet, F. Schoenahl, and F. Kraeber-Bodéré, "Assessment of acquisition protocols for routine imaging of y-90 using pet/ct," *EJNMMI research*, vol. 3, no. 1, pp. 1–12, 2013.
- [16] T. Carlier, K. P. Willowson, E. Fourkal, D. L. Bailey, M. Doss, and M. Conti, "90y-pet imaging : Exploring limitations and accuracy under conditions of low counts and high random fraction," *Medical physics*, vol. 42, no. 7, pp. 4295–4309, 2015.
- [17] H. Carrillo, M. Millardet, T. Carlier, and D. Mateus, "Low-count pet image reconstruction with bayesian inference over a deep prior," in *Medical Imaging 2021 : Image Processing*, vol. 11596. SPIE, 2021, pp. 227–235.
- [18] P. Cascarano, G. Franchini, F. Porta, and A. Sebastiani, "On the first-order optimization methods in deep image prior," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 7, no. 4, p. 041002, 2022.
- [19] P. Chakrabarty and S. Maji, "The spectral bias of the deep image prior," *arXiv preprint arXiv :1912.08905*, 2019.
- [20] R. Chandra and A. Rahmim, *Nuclear medicine physics : the basics*. Lippincott Williams & Wilkins, 2017.
- [21] Y. Chen and H. An, "Attenuation correction of pet/irm imaging," *Magnetic Resonance Imaging Clinics*, vol. 25, no. 2, pp. 245–255, 2017.
- [22] Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon, "A bayesian perspective on the deep image prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5443–5451.
- [23] C. A. Cocosco, "Brainweb : Online interface to a 3d mri simulated brain database." (*No Title*), 1997.
- [24] J. Cui, K. Gong, N. Guo, S. Wollenweber, F. Jansen, H. Liu, and Q. Li, "Sure-based stopping strategy for fine-tunable supervised pet image denoising," in *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. IEEE, 2021, pp. 1–3.
- [25] J. Cui, K. Gong, N. Guo, C. Wu, X. Meng, K. Kim, K. Zheng, Z. Wu, L. Fu, B. Xu, *et al.*, "Pet image denoising using unsupervised deep learning," *European journal of nuclear medicine and molecular imaging*, vol. 46, pp. 2780–2789, 2019.
- [26] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE transactions on medical imaging*, vol. 14, no. 1, pp. 132–137, 1995.
- [27] A. R. De Pierro and M. B. Yamagishi, "Fast em-like methods for maximum" a posteriori" estimates in emission tomography," *IEEE transactions on medical imaging*, vol. 20, no. 4, pp. 280–288, 2001.
- [28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words : Transformers for image recognition at scale," *arXiv preprint arXiv :2010.11929*, 2020.
- [29] M. J. Ehrhardt, P. Markiewicz, M. Liljeroth, A. Barnes, V. Kolehmainen, J. S. Duncan, L. Pizarro, D. Atkinson, B. F. Hutton, S. Ourselin, *et al.*, "Pet reconstruction with an anatomical mri prior using parallel level sets," *IEEE transactions on medical imaging*, vol. 35, no. 9, pp. 2189–2199, 2016.

- [30] K. Erlandsson, D. Visvikis, W. Waddington, I. Cullum, P. Jarritt, and L. Polowsky, “Low-statistics reconstruction with ab-emml,” in *2000 IEEE Nuclear Science Symposium. Conference Record (Cat. No. 00CH37149)*, vol. 2. IEEE, 2000, pp. 15–249.
- [31] R. D. Evans and R. Evans, *The atomic nucleus*. McGraw-Hill New York, 1955, vol. 582.
- [32] F. H. Fahey, “Data acquisition in pet imaging,” *Journal of nuclear medicine technology*, vol. 30, no. 2, pp. 39–49, 2002.
- [33] M. Filipović, C. Comtat, and S. Stute, “Time-of-flight (tof) implementation for pet reconstruction in practice,” *Physics in Medicine & Biology*, vol. 64, no. 23, p. 23NT01, 2019.
- [34] S. Geman, “Statistical methods for tomographic image restoration,” *Bull. Internat. Statist. Inst.*, vol. 52, pp. 5–21, 1987.
- [35] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.
- [36] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.
- [37] K. Gong, C. Catana, J. Qi, and Q. Li, “Pet image reconstruction using deep image prior,” *IEEE transactions on medical imaging*, vol. 38, no. 7, pp. 1655–1665, 2019.
- [38] —, “Direct reconstruction of linear parametric images from dynamic pet using nonlocal deep image prior,” *IEEE Transactions on Medical Imaging*, vol. 41, no. 3, pp. 680–689, 2021.
- [39] K. Gong, D. Wu, K. Kim, J. Yang, T. Sun, G. El Fakhri, Y. Seo, and Q. Li, “Mapem-net : an unrolled neural network for fully 3d pet image reconstruction,” in *15th International meeting on fully three-dimensional image reconstruction in radiology and nuclear medicine*, vol. 11072. SPIE, 2019, pp. 109–113.
- [40] P. J. Green, “Bayesian reconstructions from emission tomography data using a modified em algorithm,” *IEEE transactions on medical imaging*, vol. 9, no. 1, pp. 84–93, 1990.
- [41] S. Guo, Y. Sheng, L. Chai, and J. Zhang, “Pet image reconstruction with kernel and kernel space composite regularizer,” *IEEE Transactions on Medical Imaging*, 2023.
- [42] I. Häggström, C. R. Schmidlein, G. Campanella, and T. J. Fuchs, “Deeppet : A deep encoder–decoder network for directly solving the pet image reconstruction inverse problem,” *Medical image analysis*, vol. 54, pp. 253–262, 2019.
- [43] F. Hashimoto, H. Ohba, K. Ote, A. Kakimoto, H. Tsukada, and Y. Ouchi, “4d deep image prior : dynamic pet image denoising using an unsupervised four-dimensional branch convolutional neural network,” *Physics in Medicine & Biology*, vol. 66, no. 1, p. 015006, 2021.
- [44] F. HASHIMOTO, H. OHBA, K. OTE, and A. TERAMOTO, “Unsupervised dynamic pet image denoising with anatomical information,” *Medical Imaging and Information Sciences*, vol. 37, no. 3, pp. 58–61, 2020.
- [45] F. Hashimoto, H. Ohba, K. Ote, A. Teramoto, and H. Tsukada, “Dynamic pet image denoising using deep convolutional neural networks without prior training datasets,” *IEEE access*, vol. 7, pp. 96 594–96 603, 2019.
- [46] F. Hashimoto, Y. Onishi, K. Ote, H. Tashima, and T. Yamaya, “Fully 3d implementation of the end-to-end deep image prior-based pet image reconstruction using block iterative algorithm,” *Physics in Medicine & Biology*, vol. 68, no. 15, p. 155009, 2023.

- [47] F. Hashimoto, K. Ote, and Y. Onishi, “Pet image reconstruction incorporating deep image prior and a forward projection model,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 6, no. 8, pp. 841–846, 2022.
- [48] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, “Swin unetr : Swin transformers for semantic segmentation of brain tumors in mri images,” in *International MICCAI Brainlesion Workshop*. Springer, 2021, pp. 272–284.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers : Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [50] R. Heckel and P. Hand, “Deep decoder : Concise image representations from untrained non-convolutional networks,” *arXiv preprint arXiv :1810.03982*, 2018.
- [51] R. Heckel and M. Soltanolkotabi, “Denoising and regularization via exploiting the structural bias of convolutional generators,” *arXiv preprint arXiv :1910.14634*, 2019.
- [52] H. M. Hudson and R. S. Larkin, “Accelerated image reconstruction using ordered subsets of projection data,” *IEEE transactions on medical imaging*, vol. 13, no. 4, pp. 601–609, 1994.
- [53] C. Hui, Y. Sheng, D. Xiong, and L. Chai, “Pet image reconstruction using deep image prior and graph laplacian regularization,” in *2022 34th Chinese Control and Decision Conference (CCDC)*. IEEE, 2022, pp. 1376–1381.
- [54] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, “A fast algorithm to calculate the exact radiological path through a pixel or voxel space,” *Journal of computing and information technology*, vol. 6, no. 1, pp. 89–94, 1998.
- [55] S. P. Kim, C. Cohalan, N. Kopek, and S. A. Enger, “A guide to 90y radioembolization and its dosimetry,” *Physica Medica*, vol. 68, pp. 132–145, 2019.
- [56] D. P. Kingma and J. Ba, “Adam : A method for stochastic optimization,” *arXiv preprint arXiv :1412.6980*, 2014.
- [57] G. Konstantinou, P. Lecoq, J. M. Benloch, and A. J. Gonzalez, “Metascintillators for ultrafast gamma detectors : A review of current state and future perspectives,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 6, no. 1, pp. 5–15, 2021.
- [58] T. Li, Z. Xie, W. Qi, E. ASMA, and J. Qi, “Unsupervised pet image denoising using double over-parameterization,” 2022.
- [59] H. Lim, Y. K. Dewaraja, and J. A. Fessler, “A pet reconstruction formulation that enforces non-negativity in projection space for bias reduction in y-90 imaging,” *Physics in Medicine & Biology*, vol. 63, no. 3, p. 035042, 2018.
- [60] A. Mehranian and A. J. Reader, “Model-based deep learning pet image reconstruction using forward–backward splitting expectation–maximization,” *IEEE transactions on radiation and plasma medical sciences*, vol. 5, no. 1, pp. 54–64, 2020.
- [61] T. Merlin, S. Stute, D. Benoit, J. Bert, T. Carlier, C. Comtat, M. Filipovic, F. Lamare, and D. Visvikis, “Castor : a generic data organization and processing code framework for multi-modal and multi-dimensional tomographic reconstruction,” *Physics in Medicine & Biology*, vol. 63, no. 18, p. 185005, 2018.
- [62] M. Millardet, S. Moussaoui, J. Idier, D. Mateus, M. Conti, C. Bailly, S. Stute, and T. Carlier, “A multiobjective comparative analysis of reconstruction algorithms in the context of low-statistics 90 y-pet imaging,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 6, no. 6, pp. 629–640, 2021.

- [63] W. W. Moses, "Fundamental limits of spatial resolution in pet," *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 648, pp. S236–S240, 2011.
- [64] E. Ü. Mumcuoglu, R. M. Leahy, and S. R. Cherry, "Bayesian reconstruction of pet images : methodology and performance analysis," *Physics in Medicine & Biology*, vol. 41, no. 9, p. 1777, 1996.
- [65] J. Nuyts, D. Beque, P. Dupont, and L. Mortelmans, "A concave prior penalizing relative differences for maximum-a-posteriori reconstruction in emission tomography," *IEEE Transactions on nuclear science*, vol. 49, no. 1, pp. 56–60, 2002.
- [66] J. Nuyts, S. Stroobants, P. Dupont, S. Vleugels, P. Flamen, and L. Mortelmans, "Reducing loss of image quality because of the attenuation artifact in uncorrected pet whole-body images," *Journal of Nuclear Medicine*, vol. 43, no. 8, pp. 1054–1062, 2002.
- [67] Y. Onishi, F. Hashimoto, K. Ote, K. Matsubara, and M. Ibaraki, "Self-supervised pre-training for deep image prior-based robust pet image denoising," *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2023.
- [68] Y. Onishi, F. Hashimoto, K. Ote, H. Ohba, R. Ota, E. Yoshikawa, and Y. Ouchi, "Anatomical-guided attention enhances unsupervised pet image denoising performance," *Medical Image Analysis*, vol. 74, p. 102226, 2021.
- [69] K. Ote, F. Hashimoto, Y. Onishi, T. Isobe, and Y. Ouchi, "List-mode pet image reconstruction using deep image prior," *IEEE Transactions on Medical Imaging*, 2023.
- [70] V. Panin, M. Chen, and C. Michel, "Simultaneous update iterative algorithm for variance reduction on random coincidences in pet," in *2007 IEEE Nuclear Science Symposium Conference Record*, vol. 4. IEEE, 2007, pp. 2807–2811.
- [71] J. Qi and R. M. Leahy, "Iterative reconstruction techniques in emission computed tomography," *Physics in Medicine & Biology*, vol. 51, no. 15, p. R541, 2006.
- [72] —, "Iterative reconstruction techniques in emission computed tomography," *Phys. Med. Biol.*, vol. 51, no. 15, pp. R541–78, 2006.
- [73] A. Rahmim, J. Qi, and V. Sossi, "Resolution modeling in pet imaging : theory, practice, benefits, and pitfalls," *Medical physics*, vol. 40, no. 6Part1, p. 064301, 2013.
- [74] A. J. Reader, G. Corda, A. Mehranian, C. da Costa-Luis, S. Ellis, and J. A. Schnabel, "Deep learning for pet image reconstruction," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 1, pp. 1–25, 2020.
- [75] O. Ronneberger, P. Fischer, and T. Brox, "U-net : Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015 : 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [76] G. B. Saha, *Basics of PET imaging : physics, chemistry, and regulations*. Springer, 2015.
- [77] D. R. Schaart, G. Schramm, J. Nuyts, and S. Surti, "Time of flight in perspective : instrumental and computational aspects of time resolution in positron emission tomography," *IEEE transactions on radiation and plasma medical sciences*, vol. 5, no. 5, pp. 598–618, 2021.
- [78] G. Schramm, M. Holler, A. Rezaei, K. Vunckx, F. Knoll, K. Bredies, F. Boada, and J. Nuyts, "Evaluation of parallel level sets and bowsher's method as segmentation-free anatomical priors for time-of-flight pet reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 2, pp. 590–603, 2017.

- [79] F. Shamshad, S. Khan, S. W. Zamir, M. H. Khan, M. Hayat, F. S. Khan, and H. Fu, "Transformers in medical imaging : A survey," *Medical Image Analysis*, p. 102802, 2023.
- [80] Q. Shan, J. Wang, and D. Liu, "Deep image prior based pet reconstruction from partial data," *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2023.
- [81] C. Shen, W. Xia, H. Ye, M. Hou, H. Chen, Y. Liu, J. Zhou, and Y. Zhang, "Unsupervised bayesian pet reconstruction," *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2022.
- [82] L. A. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE transactions on medical imaging*, vol. 1, no. 2, pp. 113–122, 1982.
- [83] Z. Shi, P. Mettes, S. Maji, and C. G. Snoek, "On measuring and controlling the spectral bias of the deep image prior," *International Journal of Computer Vision*, vol. 130, no. 4, pp. 885–908, 2022.
- [84] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional ct array," *Medical physics*, vol. 12, no. 2, pp. 252–255, 1985.
- [85] A. Sitek and A. M. Celler, "Limitations of poisson statistics in describing radioactive decay," *Physica Medica*, vol. 31, no. 8, pp. 1105–1107, 2015.
- [86] C. W. Stearns, D. L. McDaniel, S. G. Kohlmyer, P. R. Arul, B. P. Geiser, and V. Shanmugam, "Rr," in *2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No. 03CH37515)*, vol. 5. IEEE, 2003, pp. 3067–3069.
- [87] S. Stute, C. Tauber, C. Leroy, M. Bottlaender, V. Brulon, and C. Comtat, "Analytical simulations of dynamic pet scans with realistic count rates properties," in *2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. IEEE, 2015, pp. 1–3.
- [88] H. Sun, L. Peng, H. Zhang, Y. He, S. Cao, and L. Lu, "Dynamic pet image denoising using deep image prior combined with regularization by denoising," *IEEE Access*, vol. 9, pp. 52 378–52 392, 2021.
- [89] F. Sureau, M. Latreche, M. Savanier, and C. Comtat, "Convergent admm plug and play pet image reconstruction," *arXiv preprint arXiv :2310.04299*, 2023.
- [90] M. Tölle, M.-H. Laves, and A. Schlaefler, "A mean-field variational inference approach to deep image prior for inverse problems in medical imaging," in *Medical Imaging with Deep Learning*. PMLR, 2021, pp. 745–760.
- [91] N. Tomic, C. J. Thompson, and M. E. Casey, "Investigation of the" block effect" on spatial resolution in pet detectors," *IEEE transactions on nuclear science*, vol. 52, no. 3, pp. 599–605, 2005.
- [92] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *arXiv :1711.10925*, 2017.
- [93] K. Van Slambrouck, S. Stute, C. Comtat, M. Sibomana, F. H. van Velden, R. Boellaard, and J. Nuyts, "Bias reduction for low-statistics pet : maximum likelihood reconstruction with a modified poisson distribution," *IEEE Transactions on medical imaging*, vol. 34, no. 1, pp. 126–136, 2014.
- [94] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [95] S. V. Venkatakrisnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE global conference on signal and information processing*. IEEE, 2013, pp. 945–948.

- [96] K. Vunckx and J. Nuyts, "Heuristic modification of an anatomical markov prior improves its performance," in *IEEE Nuclear Science Symposium & Medical Imaging Conference*. IEEE, 2010, pp. 3262–3266.
- [97] G. Wang and J. Qi, "Penalized likelihood pet image reconstruction using patch-based edge-preserving regularization," *IEEE transactions on medical imaging*, vol. 31, no. 12, pp. 2194–2204, 2012.
- [98] —, "Pet image reconstruction using kernel method," *IEEE transactions on medical imaging*, vol. 34, no. 1, pp. 61–71, 2014.
- [99] H. Wang, T. Li, Z. Zhuang, T. Chen, H. Liang, and J. Sun, "Early stopping for deep image prior," *arXiv preprint arXiv :2112.06074*, 2021.
- [100] X. Wang, Y. Wang, H. Jiang, and H. Ye, "Enhanced pet image reconstruction using deep image prior," 2021.
- [101] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment : from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [102] C. C. Watson, "New, faster, image-based scatter correction for 3d pet," *IEEE Transactions on Nuclear Science*, vol. 47, no. 4, pp. 1587–1594, 2000.
- [103] C. C. Watson, M. Casey, C. Michel, and B. Bendriem, "Advances in scatter correction for 3d pet/ct," in *IEEE Symposium Conference Record Nuclear Science 2004.*, vol. 5. IEEE, 2004, pp. 3008–3012.
- [104] B. Wohlberg, "Admm penalty parameter selection by residual balancing," *arXiv preprint arXiv :1704.06209*, 2017.
- [105] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [106] Z. Xu, M. Figueiredo, and T. Goldstein, "Adaptive admm with spectral penalty parameter selection," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 718–727.
- [107] T. Yokota, K. Kawai, M. Sakata, Y. Kimura, and H. Hontani, "Dynamic pet image reconstruction using nonnegative matrix factorization incorporated with deep image prior," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3126–3135.
- [108] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778 : L-bfgs-b : Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on mathematical software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

Annexes

A Liste des publications et présentations scientifiques

A.1 Liste des publications scientifiques

- Article soumis au journal *IEEE transactions on medical imaging* :
A. Merasli, T. Carlier, T. Liu, D. Mateus, M. Millardet, S. Moussaoui and S. Stute, “DNA : Improved Deep Image Prior Based PET Reconstruction With a Non-Negativity Constraint in Projection Space”.
La version soumise de cet article est disponible dans ce manuscrit.
- Présentation du travail sur le DNA et l’étude des hyperparamètres, avec des résultats sur la simulation 2D en ^{90}Y :
 - Oral au congrès national RITS (Recherche en Imagerie et Technologie pour la Santé, organisé par la SFGBM) en mai 2022,
 - Poster au congrès international IEEE MIC (Medical Imaging Conference, organisé par IEEE) en novembre 2022.
- Acte de conférence publié après le MIC 2022 :
A. Merasli, T. Carlier, T. Liu, D. Mateus, M. Millardet, S. Moussaoui and S. Stute, “Nested ADMM for PET reconstruction with two constraints : Deep Image Prior and non-negativity in projection space,” in 2022 IEEE Nuclear Science Symposium and Medical Imaging Conference, Proceedings (NSS/MIC). IEEE, 2022.
- Présentation orale au congrès international IEEE MIC en novembre 2023 :
A. Merasli, T. Carlier, D. Mateus, S. Moussaoui and S. Stute, “The influence of input and skip connections in PET reconstruction with Deep Image Prior,” in 2023 IEEE Nuclear Science Symposium and Medical Imaging Conference, Oral presentation (NSS/MIC). IEEE, 2023.

B Optimisation mathématique

B.1 Algorithme ADMM

B.1.1 Lagrangien

Partons du problème d’optimisation d’une fonction f , avec contrainte de linéarité :

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to } Ax = b \end{cases} \quad (\text{iii})$$

avec $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convexe.

Pour résoudre ce problème d’optimisation, on définit le Lagrangien associé :

$$L(x, y) = f(x) + y^T(Ax - b) \quad (\text{iv})$$

avec y la variable duale, aussi appelée multiplicateur de Lagrange.

Le Lagrangien vérifie les propriétés suivantes :

Problème primal associé à (iii)

$$(iii) \iff \min_{x \in \mathbb{R}^n} \sup_{y \in \mathbb{R}^m} L(x, y) \quad (v)$$

Problème dual associé à (iii), si la dualité forte est vérifiée, c'est-à-dire si le problème d'optimisation est convexe, et si un point vérifiant la contrainte $Ax = b$ existe :

$$(iii) \iff \max_{y \in \mathbb{R}^m} \inf_{x \in \mathbb{R}^n} L(x, y) = \max_{y \in \mathbb{R}^m} g(y) \quad (vi)$$

Le Lagrangien est très utile car il permet de réécrire le problème d'optimisation (iii) sans contrainte, et permet d'élargir le problème d'optimisation en introduisant la variable duale y , ce qui permet de le résoudre plus facilement.

De plus, à l'optimum x^* , le gradient du Lagrangien doit être nul :

$$\frac{\partial L(x^*, \cdot)}{\partial x} = Ax - b = 0 \quad (vii)$$

$$\frac{\partial L(\cdot, y^*)}{\partial y} = \frac{\partial f(x^*)}{\partial x} + A^T y^* = 0 \quad (viii)$$

La première équation (vii) s'appelle la faisabilité primale, qui correspond en fait au respect de la contrainte dans xvi à l'optimum x^* . La faisabilité duale correspond à la deuxième équation (viii).

B.1.2 Algorithme de montée de gradient duale

L'algorithme de montée de gradient duale (dual ascent en anglais) a pour but d'optimiser un Lagrangien, qui est une fonction de deux variables dépendant des variables primale x et duale y . Il cherche l'optimum en découplant l'optimisation :

- D'abord on cherche un optimum $x^* = \arg \min_{x \in \mathbb{R}^n} L(x, y^*)$, pour $y = y^*$ fixé,
- Puis on cherche un optimum $y^* = \arg \max_{y \in \mathbb{R}^p} L(x^*, y)$, pour $x = x^*$ fixé, à partir d'une montée de gradient.

Les itérés sont donc obtenus par la double mise à jour suivante :

$$\begin{cases} x^{k+1} := \arg \min_{x \in \mathbb{R}^n} L(x, y^k) \\ y^{k+1} := y^k + \alpha^k (Ax^{k+1} - b) \end{cases} \quad (ix)$$

Cette méthode est particulièrement avantageuse pour une fonction f qui est séparable en ses variables, c'est-à-dire pouvant s'écrire sous la forme :

$$f(x) = \sum_{i=1}^n f_i(x_i) \quad (x)$$

avec $f_i : x_i \in \mathbb{R}^{n_i} \rightarrow \mathbb{R}$

Après calculs, l'algorithme de descente :

$$\begin{cases} x_i^{k+1} := \arg \min_{x_i \in \mathbb{R}^{n_i}} L_i(x_i, y^k) \\ y^{k+1} := y^k + \alpha^k (Ax^{k+1} - b) \end{cases} \quad (xi)$$

L'avantage de cette méthode est de pouvoir paralléliser les recherches de minimum de tous les x_i .

B.1.3 Lagrangien augmenté

Pour résoudre le problème d'optimisation (iii), on peut ajouter un terme $\frac{\rho}{2}\|Ax - b\|_2^2$ qui est nul quand la contrainte est vérifiée, ce qui donne l'équivalence suivante :

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2}\|Ax - b\|_2^2 \\ \text{subject to } Ax = b \end{cases} \iff (iii) \quad (xii)$$

avec $\rho \in \mathbb{R}$ appelé paramètre de pénalité. On définit alors le Lagrangien augmenté à partir du Lagrangien (iv) :

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2 \quad (xiii)$$

$$L_\rho(x, y) = L(x, y) + \frac{\rho}{2}\|Ax - b\|_2^2 \quad (xiv)$$

En appliquant un algorithme de montée de gradient sur le Lagrangien augmenté, on obtient ce qu'on appelle la méthode des multiplicateurs :

$$\begin{cases} x^{k+1} := \arg \min_{x \in \mathbb{R}^n} L_\rho(x, y^k) \\ y^{k+1} := y^k + \rho(Ax^{k+1} - b) \end{cases} \quad (xv)$$

Le pas d'optimisation dans (xv) est habituellement choisi fixe au cours des itérations, et égal au paramètre de pénalité ρ pour que la faisabilité duale soit vérifiée [12]. Cette méthode converge dès lors que la dualité forte est vérifiée [12].

B.1.4 ADMM

L'algorithme Alternating Direction Method of Multipliers (ADMM) [12] est un algorithme pour résoudre de façon itérative un problème d'optimisation de deux variables indépendantes x et z , originellement avec une contrainte linéaire d'égalité. Il a été conçu pour tirer le meilleur de la décomposition duale, tout en gardant les propriétés de convergence du Lagrangien augmenté. Voici le problème mathématique qu'ADMM cherche à résoudre :

$$\begin{cases} \min_{x, z} f(x) + g(z) \\ \text{subject to } Ax + Bz = c \end{cases} \quad (xvi)$$

avec $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $c \in \mathbb{R}^p$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, f et g convexe.

Le Lagrangien augmenté associé à ce problème s'écrit :

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \quad (xvii)$$

La méthode des multiplicateurs résout le problème d'optimisation (xvi) avec les équations suivantes :

$$\begin{cases} (x^{k+1}, z^{k+1}) := \arg \min_{x, z} L_\rho(x, z, y^k) \\ y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{cases} \quad (xviii)$$

On remarque si f ou g est séparable, alors la décomposition duale n'est pas possible, car les variables x et z sont couplées par le terme de norme présent dans le Lagrangien augmenté.

L'idée de l'algorithme ADMM est alors de découpler ces deux variables en optimisant d'abord le Lagrangien augmenté à z et y fixés, puis à x et y fixés, comme présenté dans les équations suivantes :

$$\begin{cases} x^{k+1} := \arg \min_x L_\rho(x, z^k, y^k) \\ z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{cases} \quad (xix)$$

Dans ce cas, si une des fonctions f ou g est séparable, la décomposition duale est alors possible car les variables x et z sont maintenant découplées.

En pratique, il peut être utile de définir le résidu $r = Ax + Bz - c$ et la variable duale mise à l'échelle $u = \frac{1}{\rho}y$. Le Lagrangien augmenté en forme mise à l'échelle s'écrit alors :

$$L_\rho(x, z, y) = f(x) + g(z) + \frac{\rho}{2}\|r + \mu\|^2 - \frac{\rho}{2}\|\mu\|^2 \quad (\text{xx})$$

On en déduit l'écriture de l'algorithme ADMM en forme mise à l'échelle :

$$\begin{cases} x^{k+1} := \arg \min_x f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|_2^2 \\ z^{k+1} := \arg \min_z g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2 \\ u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c \end{cases} \quad (\text{xxi})$$

Nous utiliserons par la suite cette forme de l'algorithme ADMM.

La convergence de l'algorithme ADMM est démontrée sous certaines conditions sur les fonctions f et g et sur le Lagrangien augmenté [12].

B.1.5 Résidus associés au problème d'optimisation

Dans cette partie, nous allons calculer les résidus associés au problème d'optimisation issu de l'algorithme ADMM-Reg (Partie 1.2.5.6.4, équation (3.4)) que nous réécrivons ici :

$$\begin{cases} \arg \max_{x,v} y^T \ln(v+r) - 1^T(v+r) + g(v+r) \\ \text{subject to } Ax = v \end{cases} \quad (\text{xxii})$$

Le Lagrangien associé à ce problème d'optimisation s'écrit comme :

$$L(x, v, \lambda) = 1^T(v + \bar{b}) - y^T \ln(v + \bar{b}) + g(v + \bar{b}) + \lambda(Ax - v) \quad (\text{xxiii})$$

avec λ la variable duale associé au problème d'optimisation (xxii).

Ce problème d'optimisation est résolu par ADMM par les équations d'ADMM-Reg :

$$x^{k+1} = \arg \min_x \quad \frac{\alpha}{2}\|Ax - v^k + u^k\|_2^2 \quad (\text{xxiva})$$

$$v^{k+1} = \arg \min_v \quad 1^T(v + \bar{b}) - y^T \ln(v + \bar{b}) + g(v + \bar{b}) + \frac{\alpha}{2}\|Ax^{k+1} - v + u^k\|_2^2 \quad (\text{xxivb})$$

$$u^{k+1} = \quad u^k + Ax^{k+1} - v^{k+1} \quad (\text{xxivc})$$

Pour le problème d'optimisation 3.4, les résidus sont définis à chaque itération comme l'écart d'un terme devant être nul à l'optimum par rapport à zéro. Ils s'appuient sur les conditions de faisabilités primale et duale, qui expriment des quantités devant être nulles à l'optimum.

Le résidu primal s'appuie sur le fait que les itérés doivent respecter la contrainte du problème d'optimisation (3.4). Pour (3.4), le résidu primal à l'itération $k+1$ est défini par $r^{k+1} = Ax^{k+1} - v^{k+1}$.

[104] montrent que la faisabilité duale par rapport à la variable v est toujours vérifiée par les itérés (v^{k+1}, u^{k+1}) .

Quant à la faisabilité duale liée à la variable x , elle s'obtient en annulant la dérivée partielle du Lagrangien par rapport à x .

$$\frac{\partial L(x^*, v^*, u^*)}{\partial x} = 0 \iff A^T u^* = 0 \quad (\text{xxv})$$

Or soit h_x^{k+1} la fonction de coût de l'équation(xxiva). On a $\frac{\partial h_x^{k+1}(x^{k+1})}{\partial x} = 0$ car x^{k+1} minimise h_x^{k+1} par définition. On obtient :

$$\alpha A^T(Ax^{k+1} - v^k + u^k) = 0 \iff \alpha A^T(v^{k+1} - v^k) + A^T u^k = 0 \iff \alpha A^T u^k - s^{k+1} = 0 \quad (\text{xxvi})$$

On reconnaît le terme issu de la faisabilité duale (xxv), et on définit alors le résidu dual à l'itération $k+1$ par $s^{k+1} = -A^T(v^{k+1} - v^k)$ [104].

B.2 Calcul de pas analytique pour la mise à jour de x dans l'algorithme DNA

Pour simplifier la notation, nous omettrons l'écriture des exposants $n+1$ (indice des itérations externes) et notons $\nabla_x h(x^{n+1,k}) = \nabla_x h(x^k) = d^k$. Pour calculer le pas optimal $\zeta^{n+1,k}$ à utiliser dans l'équation (2.17), nous pouvons effectuer une recherche linéaire en minimisant $h(x^k - \zeta d^k)$:

$$h(x^k - \zeta d^k) \tag{xxvii}$$

$$= \frac{\rho_1}{2} \|x^k - f(\theta^n|z) + \mu^n - \zeta d^k\|_2^2 + \frac{\rho_2}{2} \|Ax^k - v^k + u^k - \zeta Ad^k\|_2^2, \tag{xxviii}$$

$$= \frac{\rho_1}{2} \|x^k - f(\theta^n|z) + \mu^n\|_2^2 + \frac{\rho_1}{2} \zeta^2 \|d^k\|_2^2 - 2\zeta (d^k)^T \frac{\rho_1}{2} (x^k - f(\theta^n|z) + \mu^n) + \frac{\rho_2}{2} \|Ax^k - v^k + u^k\|_2^2 + \frac{\rho_2}{2} \zeta^2 \|Ad^k\|_2^2 - 2\zeta (d^k)^T A^T \frac{\rho_2}{2} (Ax^k - v^k + u^k), \tag{xxix}$$

$$= \frac{\rho_1}{2} \|x^k - f(\theta^n|z) + \mu^n\|_2^2 + \frac{\rho_1}{2} \zeta^2 \|d^k\|_2^2 + \frac{\rho_2}{2} \|Ax^k - v^k + u^k\|_2^2 + \frac{\rho_2}{2} \zeta^2 \|Ad^k\|_2^2 - \zeta (d^k)^T d^k. \tag{xxx}$$

Finalement nous obtenons :

$$h(x^k - \zeta d^k) = \zeta^2 \left(\frac{\rho_1}{2} \|d^k\|_2^2 + \frac{\rho_2}{2} \|Ad^k\|_2^2 \right) - \zeta \|d^k\|_2^2 + \left(\frac{\rho_1}{2} \|x^k - f(\theta^n|z) + \mu^n\|_2^2 + \frac{\rho_2}{2} \|Ax^k - v^k + u^k\|_2^2 \right). \tag{xxxii}$$

Le minimum de ce trinôme en ζ est atteint pour la valeur :

$$\zeta^{n+1,k} = \frac{-(-\|d^k\|_2^2)}{2 \left(\frac{\rho_1}{2} \|d^k\|_2^2 + \frac{\rho_2}{2} \|Ad^k\|_2^2 \right)}, \tag{xxxiii}$$

$$= \frac{\|\nabla_x h(x^{n+1,k})\|^2}{\rho_2 \|A \nabla_x h(x^{n+1,k})\|^2 + \rho_1 \|\nabla_x h(x^{n+1,k})\|^2}, \tag{xxxiiii}$$

qui est le résultat proposé dans l'équation (2.19).

B.3 Dérivation plus complète de l'algorithme ADMM-Reg

Cette annexe est rédigée pour ajouter quelques détails mathématiques et précisions sur l'algorithme de reconstruction utilisé pour résoudre les problèmes d'optimisation (1.51b) utilisé dans ADMM-Reg et (2.16b) utilisé dans le DNA. Les équations de cette sous-section sont les mêmes que celles présentées dans le papier de Lim et al. [59], avec de légères modifications sur certains symboles de comparaison.

B.3.1 Problème d'optimisation initial

Nous démarrons cette annexe avec le problème d'optimisation suivant :

$$\min_x L(y|x) + R(x) + g(\bar{y}, y), \tag{xxxv}$$

avec

$$g(\bar{y}, y) = \sum_i g_i(\bar{y}_i, y_i), \quad g_i(\bar{y}_i, y_i) = \begin{cases} \infty, & \bar{y}_i < 0 \\ \infty, & \bar{y}_i = 0, y_i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{xxxvi}$$

La fonction barrière g est définie ici comme une fonction bivariable des variables \bar{y} et y . Cette contrainte est nécessaire pour pouvoir définir le terme $\ln(\bar{y})$ dans la fonction de vraisemblance.

La dérivation est alors la même que dans [59], en introduisant la contrainte $Ax = v$, et en résolvant ce problème sous contrainte avec l'algorithme ADMM sous forme mise à l'échelle [12] :

$$\begin{aligned}
x^{n+1,k+1} &= \arg \min_x h_x(x) \\
\text{with } h_x(x) &= \frac{\rho_2}{2} \|Ax - v^{n+1,k} + u^{n+1,k}\|_2^2 + R(x), & (\text{xxxviiia}) \\
v^{n+1,k+1} &= \arg \min_v h_v(x) \\
\text{with } h_v(x) &= 1^T(v + \bar{b}) - y^T \ln(v + \bar{b}) + g(v + \bar{b}, y) \\
&+ \frac{\rho_2}{2} \|Ax^{n+1,k+1} - v + u^{n+1,k}\|_2^2, & (\text{xxxviiib}) \\
u^{n+1,k+1} &= u^{n+1,k} + Ax^{n+1,k+1} - v^{n+1,k+1}. & (\text{xxxviiic})
\end{aligned}$$

B.3.2 Détails sur la mise à jour du sinogramme v

Ici, nous voulons surtout souligner la dérivation analytique de la mise à jour de v , et préciser que la solution de ce problème d'optimisation (2.16b) existe toujours, ainsi que $v_i + \bar{b}_i$ est autorisé à être nul si et seulement si $y_i = 0$. Autrement, $v_i + \bar{b}_i > 0$.

Considérons la fonction $h_v^g : v \in \mathbb{R}^{\mathbb{I}} \mapsto h_v(v) - g(v + \bar{b}, y)$. Nous allons écrire la fonction h_v^g comme une somme sur chaque bin de sinogramme :

$$h_v^g(v) = \sum_i q(v_i), \quad (\text{xxxvii})$$

avec :

$$q(v_i) = \quad (\text{xxxviii})$$

$$v_i + \bar{b}_i - y_i \ln(v_i + \bar{b}_i) + \frac{\rho_2}{2} ([Ax^{n+1,k+1}]_i - v_i + u_i^{n+1,k})^2. \quad (\text{xxxix})$$

Nous avons décidé de ne pas indexer la fonction q sur le bin de sinogramme i par souci de simplicité des notations.

Premier cas : $y_i = 0$:

Si $y_i = 0$, la dérivée de la fonction q est donnée par :

$$q'(v_i) = 1 - \rho_2([Ax^{n+1,k+1}]_i - v_i + u_i^{n+1,k}), \quad (\text{xl})$$

on obtient alors directement :

$$q'(\hat{v}_i) = 0 \iff \hat{v}_i = [Ax^{n+1,k+1}]_i + u_i^{n+1,k} - \frac{1}{\rho_2}. \quad (\text{xli})$$

Or, q' est une fonction affine, donc q' est négatif pour $v_i < \hat{v}_i$ et positif sinon. Enfin, q est décroissante puis croissante, ce qui conduit à un minimum atteint en $v_i = \hat{v}_i$.

La fonction g doit être prise en compte pour vérifier les valeurs interdites de v_i . La fonction g impose la contrainte $v_i^{n+1,k+1} + \bar{b}_i \geq 0$, car $\bar{y}_i < 0$ est interdit. Pour garantir cette contrainte, une dernière étape de mise à jour a été proposée par [59] :

$$v_i^{n+1,k+1} = [\hat{v}_i + \bar{b}_i]_+ - \bar{b}_i. \quad (\text{xlii})$$

Cette étape signifie que si $v_i^{n+1,k+1}$ est inférieur à $-\bar{b}_i$, alors cette valeur n'est pas autorisée. Sachant que la fonction q est croissante pour des valeurs supérieures à \hat{v}_i , alors $v_i^{n+1,k+1}$ doit être fixé à $-\bar{b}_i$. Sinon le minimum est choisi en $v_i^{n+1,k+1} = \hat{v}_i$.

Second cas : $y_i > 0$:

La dérivée de la fonction q est donnée par l'expression suivante :

$$q'(v_i) = 1 - \frac{y_i}{v_i + \bar{b}_i} - \rho_2([Ax^{n+1,k+1}]_i - v_i + u_i^{n+1,k}). \quad (\text{xliii})$$

En mettant au même dénominateur, on obtient :

$$q'(v_i) = \frac{v_i + \bar{b}_i - y_i - \rho_2(v_i + \bar{b}_i)([Ax^{n+1,k+1}]_i - v_i + u_i^{n+1,k})}{v_i + \bar{b}_i}. \quad (\text{xliv})$$

En réarrangeant les termes, on obtient un polynôme du second degré au numérateur :

$$q'(v_i) = \rho_2 \frac{v_i^{(2)} + v_i \left(\bar{b}_i + \frac{1}{\rho_2} - [Ax^{n+1,k+1}]_i - u_i^{n+1,k} \right)}{v_i + \bar{b}_i} + \rho_2 \frac{\left(-\bar{b}_i([Ax^{n+1,k+1}]_i + u_i^{n+1,k}) + \frac{\bar{b}_i - y_i}{\rho_2} \right)}{v_i + \bar{b}_i}. \quad (\text{xlv})$$

Pour étudier les variations de la fonction q , le signe du trinôme suivant doit être étudié :

$$\begin{aligned} p(v_i) &= v_i^{(2)} + v_i \left(\bar{b}_i + \frac{1}{\rho_2} - [Ax^{n+1,k+1}]_i - u_i^{n+1,k} \right) \\ &+ \left(-\bar{b}_i([Ax^{n+1,k+1}]_i + u_i^{n+1,k}) + \frac{\bar{b}_i - y_i}{\rho_2} \right) \\ &= v_i^{(2)} + 2\kappa v_i - \gamma, \end{aligned} \quad (\text{xlvi})$$

et

$$\gamma = \bar{b}_i([Ax^{n+1,k+1}]_i + u_i^{n+1,k}) - \frac{\bar{b}_i - y_i}{\rho_2} \quad (\text{xlvii})$$

$$= \gamma = -b_i \xi_i + \frac{y_i}{\rho_2}, \quad (\text{xlviii})$$

avec $\xi_i = \frac{1}{\alpha} - [Ax^{n+1,k+1}]_i - u_i^{n+1,k}$

Calculons le discriminant Δ du trinôme p :

$$\Delta = 4(\kappa^2 + \gamma) \quad (\text{xlix})$$

$$= 4 \left(\frac{1}{4}(\bar{b}_i + \xi_i)^2 + \frac{1}{4}(-\bar{b}_i \xi_i + \frac{y_i}{\alpha}) \right) \quad (\text{l})$$

$$= (\bar{b}_i + \xi_i)^2 - 4\bar{b}_i \xi_i + 4\frac{y_i}{\alpha} \quad (\text{li})$$

$$= (\bar{b}_i - \xi_i)^2 + 4\frac{y_i}{\alpha} \quad (\text{lii})$$

$$> 0, \quad \text{car } y_i > 0. \quad (\text{liii})$$

Ce trinôme a donc deux solutions $v_i^{(1)}$ et $v_i^{(2)}$ données par :

$$\begin{cases} v_i^{(1)} = -\kappa - \sqrt{\kappa^2 + \gamma} \\ v_i^{(2)} = -\kappa + \sqrt{\kappa^2 + \gamma} \end{cases} \quad (\text{liv})$$

Or :

$$v_i^{(2)} + \bar{b}_i = -\kappa + \sqrt{\kappa^2 + \gamma} + \bar{b}_i \quad (\text{lv})$$

$$= -\kappa + \sqrt{\frac{\Delta}{4}} + \bar{b}_i \quad (\text{lvi})$$

$$= -\frac{1}{2}(\xi_i + \bar{b}_i) + \sqrt{\frac{1}{4} \left((\bar{b}_i - \xi_i)^2 + 4\frac{y_i}{\alpha} \right)} + \bar{b}_i \quad (\text{lvii})$$

$$= \frac{1}{2} \left((\bar{b}_i - \xi_i) + \sqrt{(\bar{b}_i - \xi_i)^2 + 4\frac{y_i}{\alpha}} \right) \quad (\text{lviii})$$

En utilisant l'inégalité suivante :

$$\sqrt{(\bar{b}_i - \xi_i)^2 + 4\frac{y_i}{\rho_2}} > \sqrt{(\bar{b}_i - \xi_i)^2}, \quad \text{car } y_i > 0 \quad (\text{lix})$$

$$(\text{lx})$$

et en remarquant que :

$$\sqrt{(\bar{b}_i - \xi_i)^2} = |\bar{b}_i - \xi_i| \geq \bar{b}_i - \xi_i, \quad (\text{lx})$$

on en conclut que $v_i^{(2)} + \bar{b}_i > 0$. De la même façon, on montre que $v_i^{(1)} + \bar{b}_i < 0$. Finalement :

$$v_i^{(1)} < -\bar{b}_i < v_i^{(2)} \quad (\text{lxii})$$

On ne s'intéresse qu'aux valeurs de v_i supérieures à $-\bar{b}_i$ d'après la fonction barrière g . On a donc :

$$\begin{cases} p(v_i) < 0, & \text{si } -\bar{b}_i < v_i < v_i^{(2)} \\ p(v_i) \geq 0, & \text{si } v_i \geq v_i^{(2)}, \end{cases} \quad (\text{lxiii})$$

Comme le dénominateur de q' est positif pour $v_i > -\bar{b}_i$, q' décroît jusqu'à $v_i^{(2)}$, puis croît ensuite. On en déduit que $v_i^{(2)}$ est le minimum de q , et que la contrainte imposée par la fonction barrière g est bien respectée d'après (lxii).

Remarque : dans le cas où $\kappa \geq 0$, on multiplie l'expression (liv) par son conjugué car elle est plus stable numériquement :

$$\hat{v}_i = \frac{\gamma}{\sqrt{\kappa^2 + \gamma} + \kappa} \quad (\text{lxiv})$$

La mise à jour de v résolvant (2.16b) est donnée par cette formule :

$$v_i^{n+1,k+1} = \begin{cases} [Ax^{n+1,k+1}]_i + u_i^{n+1,k} - \frac{1}{\rho_2}, & y_i = 0 \\ \sqrt{\kappa^2 + \gamma} - \kappa, & y_i > 0, \kappa < 0 \\ \frac{\gamma}{\sqrt{\kappa^2 + \gamma} + \kappa}, & y_i > 0, \kappa \geq 0 \end{cases} \quad (\text{lxv})$$

L'étape finale suivante est réalisée :

$$v_i^{n+1,k+1} = [v_i^{n+1,k+1} + \bar{b}_i]_+ - \bar{b}_i. \quad (\text{lxvi})$$

Cette étape n'est en fait utile que dans le cas $y_i = 0$. En effet, si $y_i > 0$ alors $v_i^{n+1,k+1} + \bar{b}_i > 0$.

B.3.3 Discussion sur l'article de Lim et al.

L'article [59] écrit par Lim et al. a introduit l'algorithme ADMM-Reg, d'abord sans régularisation, avec la contrainte suivante $Ax + \bar{b} > 0$ en raison de la présence du terme $\ln(Ax + \bar{b})$. Lors de l'introduction de leur fonction barrière g^{Lim} et de la régularisation, une contrainte $Ax + \bar{b} \geq 0$ a été utilisée. La conclusion de notre dérivation mathématique est que les équations de l'algorithme ADMM-Reg résolvent le problème d'optimisation (xxxiv) avec la fonction barrière définie dans (xxxv).

C Outils numériques

C.1 CASToR

CASToR (Customizable and Advanced Software for Tomographic Reconstruction) est un logiciel open-source de reconstruction tomographique (TEMP, TDM, TEP) écrit en C++ et disponible sur Gitlab (<https://gitlab.com/castor-collaboration/castor>). Ce logiciel se veut générique pour pouvoir prendre en compte facilement des extensions par un utilisateur novice (nouvelle géométrie de scanner, nouvel algorithme de reconstruction, etc.). Des reconstructions statiques ou dynamiques (plusieurs acquisitions consécutives au cours du temps), avec ou sans données TOF peuvent être réalisées.

C.1.1 Présentation détaillée du fonctionnement de CASToR

CASToR s'utilise à partir d'une ligne de commande pour lancer l'exécutable *castor-recon*. Il est alors possible de préciser (au minimum) :

- les dimensions et le nombre des voxels à reconstruire,
- l'algorithme pour discrétiser l'opérateur de projection,
- l'algorithme de reconstruction avec nombre d'itérations et sous-ensemble associé, ainsi que la pénalité à utiliser avec son paramètre de pénalité s'il y a régularisation,
- le chemin pour stocker les sorties de CASToR (images reconstruites, fichiers de log, etc.),
- le chemin pour accéder aux données brutes à reconstruire (celles-ci sont encapsulées sous un format *.cdf* lisible par CASToR. Ce fichier prend en compte les coïncidences prompts, l'atténuation, la normalisation, et l'estimation des coïncidences aléatoires et diffusées. Le calcul de ces corrections doit être effectué en amont de la reconstruction, CASToR les applique pendant la reconstruction.).

Plus spécifiquement, le programme principal *castor-recon.cc* va appeler les différentes parties nécessaires à traiter les données et effectuer la reconstruction en fonction des paramètres demandés par l'utilisateur. Les opérations propres à l'algorithme de reconstruction sont organisées séquentiellement par les fonctions du fichier *oOptimizerManager.cc* qui appellent les fonctions de l'algorithme de reconstruction (soit dans le fichier *vOptimizer.cc* pour les fonctions communes à tous les algorithmes de reconstruction, soit dans le fichier *iOptimizerADMMLim.cc* par exemple pour les fonctions spécifiques à l'algorithme de reconstruction ADMM-Reg). A chaque itération, chaque algorithme de reconstruction parcourt une fois le fichier de données et effectue des calculs lors de 8 sous-étapes :

- projection de l'image courante (fonction *DataStep1ForwardProjectModel*),
- calcul des quantités nécessitant une rétroprojection (fonctions *DataStep3BackwardProjectSensitivity*, *DataStep5ComputeCorrections* et *DataSpaceSpecificOperations*),
- calcul de certaines figures de mérites (fonction *DataStep8ComputeFOM*),
- des fonctions supplémentaires sont utilisables si besoin (fonctions *DataStep2Optional*, *DataStep4Optional* et *DataStep6Optional*).

Une fois ces 8 sous-étapes réalisées, une boucle sur les voxels de l'image reconstruite est réalisée pour mettre à jour l'image reconstruite selon l'équation de mise à jour des itérés de l'algorithme choisi (fonctions *ImageUpdateStep* et *ImageSpaceSpecificOperations*).

C.1.2 Contributions apportées au logiciel CASToR

Au cours de cette thèse, des développements ont été ajoutés au logiciel CASToR :

De nouvelles fonctionnalités ont été introduites dans la branche développement concernant les algorithmes d'optimisation :

- un mécanisme pour les algorithmes d'optimisation utilisant des sous-étapes ou des sous-itérations, tels que les algorithmes dérivés à partir d'un ADMM (ADMM-Reg, DIPRecon etc.),
- un mécanisme de traitement des données avant et après l'optimisation. Cela est utile pour toute étape de pré ou post-traitement nécessitant une boucle sur les éléments de

données,

- la possibilité de fournir à un algorithme d’optimisation un ou plusieurs fichiers supplémentaires d’entrée dans l’espace de données.

De plus, une pénalité et deux optimiseurs ont été ajoutés grâce à ces nouvelles fonctionnalités :

- l’algorithme APPGML [62] (cf. partie 1.2.5.6.3) avait été développé en interne. Le code a été repris, commenté et ajouté dans la branche principale du Gitlab,
- l’algorithme ADMM-Reg [59] (cf. partie 1.2.5.6.4),
- la pénalité quadratique (cf. partie 1.2.5.7.1).

Une extension de l’algorithme ADMM-Reg pour reconstruire des données TOF ainsi qu’effectuer une reconstruction indirecte de données dynamiques a été ajoutée.

D Résultats additionnels

D.1 Autres résultats sur l’expérience basique ”DIP ReLU”

Cette annexe montre les résultats obtenus lors de l’expérience basique DIP ReLU (cf. partie 2.5.1) pour les trois autres configurations d’une image TEP à reproduire avec le DIP utilisant une entrée anatomique.

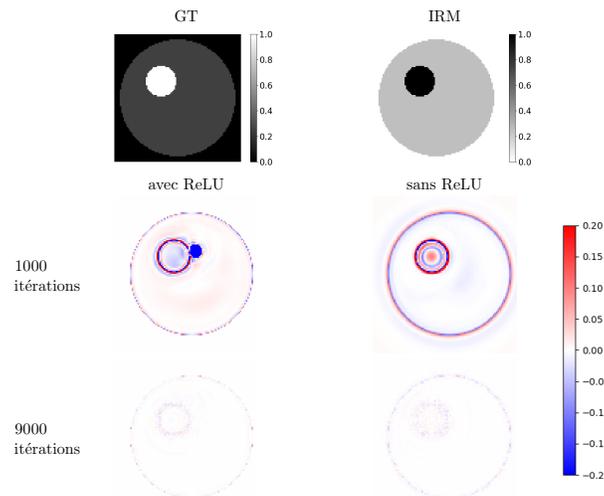


FIGURE D.1 – Expérience DIP ReLU : reproduire une image TEP (première ligne à droite) avec le DIP prenant en entrée une image anatomique (première ligne à gauche). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (colonne de gauche) et sans ReLU (colonne de droite), après 1000 itérations (deuxième ligne) et 9000 itérations (troisième ligne), lors de l’ajustement de l’image TEP avec l’insert chaud. L’image anatomique contient une zone en hypersignal.

La figure D.1 montre les images de biais obtenues pour l’image TEP avec insert chaud et l’image anatomique avec une zone en hypersignal. Les deux réseaux de neurones avec et sans ReLU ont pu ajuster le fantôme TEP avec un insert chaud après 9 000 itérations, comme le montre la dernière ligne de la figure D.1. Toutefois, lorsque la ReLU a été utilisée, certains pixels de l’image de biais forment une région biaisée négativement à 1000 itérations (région bleue dans l’image en haut à gauche de la Fig. D.1). La GT ayant une valeur de 0,2, ces pixels bleus indiquent que la sortie du réseau de neurones a produit des pixels nuls au cours des premières itérations de l’ajustement du DIP.

La figure D.2 montre les images de biais obtenues pour l’image TEP avec insert froid et l’image anatomique avec une zone en hypersignal contrairement à la figure 2.6 du chapitre 2. On compare un réseau de neurones DIP avec ou sans la fonction d’activation ReLU finale, pour 1000 et 9000 itérations. Comme dans la figure 2.6, l’insert froid n’est pas biaisé dans le cas

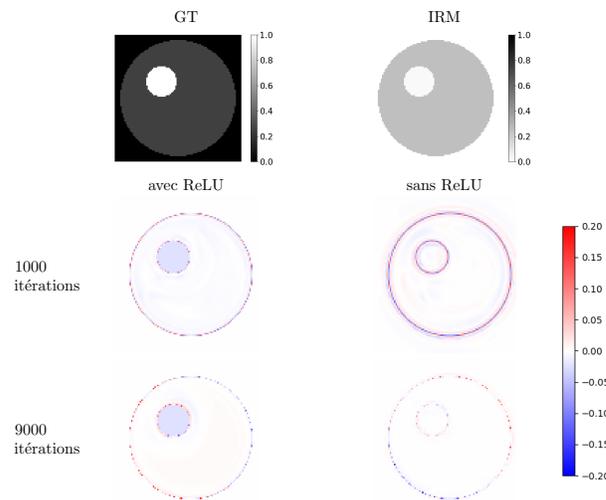


FIGURE D.2 – Expérience DIP ReLU : reproduire une image TEP (*première ligne à droite*) avec le DIP prenant en entrée une image anatomique (*première ligne à gauche*). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (*colonne de gauche*) et sans ReLU (*colonne de droite*), après 1000 itérations (*deuxième ligne*) et 9000 itérations (*troisième ligne*), lors de l’ajustement de l’image TEP avec l’insert froid. L’image anatomique contient une zone en hypersignal.

de l’architecture sans ReLU pour 1000 et 9000 itérations. Pour 1000 itérations, les contours semblent légèrement plus biaisés qu’avec une image anatomique avec la zone en hyposignal (figure 2.6 du chapitre 2). En revanche, avec ReLU, l’insert froid est bleu ce qui signifie que la ReLU biaise cette zone. En effet, la majorité des pixels ont été mis à 0 lorsque la ReLU était présente. Après 9000 itérations avec la ReLU, l’insert froid reste toujours biaisé contrairement aux résultats de la figure 2.6 où certains pixels commençaient à prendre la valeur de la GT. Nous faisons alors l’hypothèse que l’utilisation d’une image anatomique contenant une zone en hypersignal pour ajuster une image TEP avec un insert froid par le DIP est plus lent qu’une image anatomique contenant une zone en hyposignal.

La figure D.3 montre les images de biais obtenues pour l’image TEP avec insert chaud, mais l’image anatomique avec une zone en hyposignal contrairement à la figure D.1. L’architecture sans ReLU arrive à ajuster l’image TEP après 9000 itérations car l’image de biais est quasiment blanche. En revanche, certains pixels de l’image produite par l’architecture avec ReLU après 9000 itérations sont biaisés, quand d’autres ne le sont pas. Ce comportement rappelle celui de la figure 2.6 dans le cas de l’insert TEP froid, où l’architecture avec ReLU finit par supprimer petit à petit le biais dans l’insert en commençant par le centre. De plus, les images à 1000 itérations semblent plus biaisées pour les deux architectures comparés au cas de l’image anatomique avec insert chaud (cf. figure D.1). A l’inverse de l’image TEP avec insert froid, nous faisons l’hypothèse que l’utilisation d’une image anatomique contenant une zone en hypersignal permet d’ajuster l’image TEP avec un insert chaud par le DIP plus rapidement.

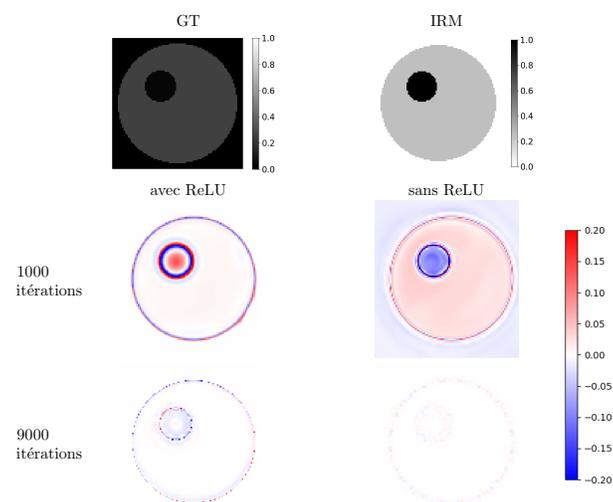


FIGURE D.3 – Expérience DIP ReLU : reproduire une image TEP (*première ligne à droite*) avec le DIP prenant en entrée une image anatomique (*première ligne à gauche*). La figure montre ensuite les images de biais entre la GT et les sorties du réseau DIP avec ReLU (*colonne de gauche*) et sans ReLU (*colonne de droite*), après 1000 itérations (*deuxième ligne*) et 9000 itérations (*troisième ligne*), lors de l’ajustement de l’image TEP avec l’insert chaud. L’image anatomique contient une zone en hyposignal.