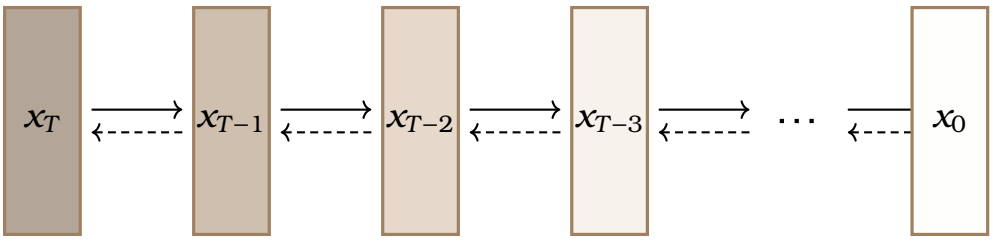
Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Compressed Sensing MRI using Score-based Implicit Model

Διπλωματικη Εργασια

του

ΔΗΜΗΤΡΙΟΥ Γ. ΔΗΜΟΥ



Επιβλέποντες: Γεώργιος Στάμου, Καθηγητής Ε.Μ.Π.
Αλέξανδρος Δημάκης, Καθηγητής UT Austin

Αθήνα, Οκτώβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστών

Compressed Sensing MRI using Score-based Implicit Model

Διπλωματικη Εργασια

του

ΔΗΜΗΤΡΙΟΥ Γ. ΔΗΜΟΥ

Επιβλέποντες: Γεώργιος Στάμου, Καθηγητής Ε.Μ.Π.
Αλέξανδρος Δημάκης, Καθηγητής UT Austin

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Νοεμβρίου 2022.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γεώργιος Στάμου	Αλέξανδρος Δημάκης	Αθανάσιος Βουλόδημος
Καθηγητής Ε.Μ.Π.	Καθηγητής UT Austin	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστών

.....
Δημήτριος Δήμος,
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

10η Νοεμβρίου 2022

Copyright © – All rights reserved. Με την επιφύλαξη παντός δικαιώματος.
Δημήτριος Δήμος, 2022.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, των Επιβλεπόντων, ή της επιτροπής που την ενέκρινε.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

Περίληψη

Σε αυτή την εργασία μελετάμε μια πρόσφατα αναδυθείσα οικογένεια παραγωγικών μοντέλων: μοντέλα diffusion-based, και προτείνουμε μία νέα μέθοδο αξιοποίησής τους για την επίλυση προβλημάτων αντιστροφής με σημαντικά μειωμένο υπολογιστικό κόστος. Την τελευταία τριετία, τα παραγωγικά μοντέλα diffusion αποτελούν σημαντικό κέντρο της έρευνας που ασχολείται με την παραγωγή δεδομένων και έχουν σημειώσει ιλιγγιώδη πρόοδο σε τομείς όπως η παραγωγή εικόνας (καθοδηγούμενη από κείμενο και μη), σχήματος και μουσικής, μεταξύ άλλων. Τα εν λόγω μοντέλα έχουν επιτύχει state-of-the-art επίδοση αναφορικά με την ποιότητα σύνθεσης εικόνων, υπερβαίνοντας μέχρι και αυτή των GANs. Ταυτόχρονα, πολλαπλά ερευνητικά έργα έχουν αξιοποιήσει τις παραγωγικές ικανότητες των μοντέλων diffusion για την προσέγγιση προϋπαρχόντων προβλημάτων και παράξει αποτέλεσματα που υποσκελίζουν προηγούμενες τεχνικές. Πιο ειδικά, τα μοντέλα score-based έχουν χρησιμοποιηθεί σε ένα ενιαίο framework προσέγγισης γραμμικών προβλημάτων αντιστροφής (linear inverse problems), το οποίο, ωστόσο, πάσχει από υψηλό υπολογιστικό κόστος που απαγορεύει την χρήση του σε προβλήματα πραγματικού κόσμου. Στα πλαίσια αυτής της διπλωματικής, εκμεταλλευόμαστε την ισοδυναμία των μοντέλων score-based και diffusion, και προτείνουμε τον αλγόριθμο Score-based Implicit Model (SBIM). Ο αλγόριθμος SBIM αποτελεί επαναπαραμετροποίηση του αλγορίθμου DDIM - ο οποίος χρησιμοποιεί diffusion μοντέλα για να παράγει δείγματα - ώστε να χρησιμοποιεί score-based μοντέλα. Με αυτόν τον τρόπο, εκμεταλλευόμαστε την εγγενή ταχύτητα του DDIM για να επεκτείνουμε το γενικό framework προσέγγισης αντίστροφων προβλημάτων με μοντέλα score-based, προτείνοντας έναν αλγόριθμο με σημαντικά χαμηλότερο υπολογιστικό κόστος. Στην εργασία παρουσιάζουμε εκτένως την μαθηματική προέλευση του αλγορίθμου SBIM και τον εφαρμόζουμε με επιτυχία στην πραγματοποίηση Compressed Sensing σε μαγνητικές τομογραφίες (MRI). Τα αποτελέσματα, βάσει των μετρικών αξιολόγησης SSIM, PSNR και χρόνου, επιβεβαιώνουν την ανταγωνιστικότητα του αλγορίθμου SBIM έναντι των προηγούμενων μεθόδων και συνιστούν μια εναλλακτική τακτική για την επιτάχυνση της επίλυσης προβλημάτων αντιστροφής με μοντέλα diffusion.

Λέξεις Κλειδιά

παραγωγικό μοντέλο, μοντέλο διάχυσης, μοντέλο score-based, εκ των υστέρων δειγματοληψία, αποθορυβοποίηση, στοχαστική διαφορική εξίσωση, πρόβλημα αντιστροφής, compressed sensing, μαγνητική τομογραφία, langevin dynamics, ancestral sampling, implicit model

Abstract

In this thesis we study a newly emerged family of generative models; diffusion-based models, and propose a new method for solving inverse problems with significantly reduced computational cost. During the last three years, generative diffusion models constitute an important pole of interest for research that focuses on data generation and have contributed in the rapid evolution of domains among which are image generation (text-guided or not), shape and music generation. These models have achieved state-of-the-art performance regarding image synthesis quality, even beating GANs. In addition, multiple research works have utilized the generative capabilities of diffusion models to approach preexisting problems and have produced results that undermine previous techniques. More specifically, score-based models have been incorporated in a unified framework that approaches linear inverse problems, but suffers from high computational cost that prohibits its real-world deployment. In this thesis, we employ the equivalence between diffusion and score-based models and propose Score-based Implicit Model (SBIM) algorithm. SBIM is a reparameterization of DDIM algorithm - which uses diffusion models by default to produce samples - to incorporate score-based models. This way, we take advantage of DDIM's innate speed in order to extend the score-based framework by proposing an algorithm with significantly lower computational cost. In this work, we demonstrate the mathematical derivation of SBIM and successfully apply it to perform Compressed Sensing in MRI. The results, based on SSIM, PSNR and time evaluation metrics, confirm that SBIM can compete against previous methods and comprises a new tactic for accelerating inverse problem solutions using diffusion models.

Keywords

generative model, diffusion mode, score-based model, a posteriori sampling, denoising, stochastic differential equation, inverse problem, compressed sensing, magnetic resonance imaging, langevin dynamics, ancestral sampling, implicit model

στη μνήμη του παππού μου, Θύμιου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω καταρχάς τους καθηγητές κ. Γεώργιο Στάμου και κ. Αλέξανδρο Δημάκη για την ευκαιρία που μου έδωσαν να εκπονήσω τη διπλωματική μου υπό την καθοδήγησή τους και για την εξαιρετική συνεργασία που είχαμε κατά τη διάρκειά της. Θα ήθελα να πω ένα μεγάλο ευχαριστώ στον Δρ. Γεώργιο Σιόλα για την σημαντική υποστήριξή του σε τεχνικά ζητήματα, καθώς και για τη πρόσβαση που μου προσέφερε στους υπολογιστικούς πόρους του ARIS HPC. Φυσικά, για αυτή την πρόσβαση ευχαριστώ και το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας (ΕΔΥΤΕ Α.Ε.), και προσωπικά τον κ. Δημήτριο Δελλή για την εξυπηρετική και άμεση τεχνική υποστήριξή του.

Ολοκληρώνοντας τις σπουδές μου καταλαβαίνω ότι οι άνθρωποι μπορούν να είναι δίπλα σου, χωρίς την φυσική τους παρουσία. Κι ένα τηλεφώνημα την εβδομάδα, αρκεί για να σε κάνει να αισθανθείς την παρουσία τους. Σε αυτή την συνειδητοποίηση με οδήγησαν οι γονείς μου, Γιώργος Δήμος και Μαρία Καρανίκα, τους οποίους θέλω να ευχαριστήσω και να τους πω ότι τους αγαπώ από τα βάθη της ψυχής μου. Μαμά, σε ευχαριστω, γιατί είσαι ο βράχος μου, το άτομο που μου αποδεικνύει επανειλημμένα πως δεν μπορεί να υπάρχει άνθρωπος που να με αγαπά περισσότερο. Και για μένα, δεν υπάρχει άνθρωπος που να αγαπώ περισσότερο. Για μένα, είσαι η βασίλισσα. Πατέρα, σε ευχαριστώ. Όσο μεγαλώνω σε βλέπω όλο και περισσότερο στον καθρέπτη, κι αυτό με χαροποιεί. Παιδί του χωριού, με μυαλό πολλές δεκαετίες μπροστά από την εποχή του. Σε αγαπώ κι είμαι περήφανος για σένα.

Σε ευχαριστώ Βασιλική, μονάκριβή μου, για πάρα πολλά πράγματα: επειδή με πρόσεχες όταν ήμουν τεσσάρων σαν να είσαι ενήλικη, ενώ ήσουν μόνο έξι, επειδή με προστάτεψες από κακιές βουλήσεις, επειδή με αγάπαγες και με αγαπάς πολύ. Να ξέρεις, πάντοτε προσπαθούσα να γίνω τόσο ικανός όσο εσύ, γιατί σε θαύμαζα - ήσουν και πρώτη, αλλά και νικήτρια. Δεν τα έχω καταφέρει, μάλλον γιατί δεν γίνεται. Είσαι ο πιο δυνατός άνθρωπος που ξέρω. Είσαι μια γλυκιά σοφία χρόνων στο σώμα κοριτσιού. Σε αγαπώ, αδερφούλα.

Πέντε χρόνια στη σχολή Ηλεκτρολόγων, δεν μπορεί να είναι βαρετά όταν έχεις φίλους σαν την Ναυσικά Αμπατζή, τον Χρήστο Δημόπουλο, τον Γιώργο Θεμελή και τη Μαρία Μουτούση. Ευχαριστώ αυτά τα παιδιά, την παρέα μου στην Αθήνα. Πέντε υπέροχα χρόνια, δίπλα σας, πέρασαν σαν μια βδομάδα. Πόσα ξενύχτια (όχι μόνο διαβάσματος), πόσες εργασίες, πόσο κώδικα... Ναυσικά, πάντα δίπλα μου άνευ όρων, με στήριξες όπως μόνο εσύ θα μπορούσες και σε ευγνωμονώ γι'αυτό. Ησουν ο άνθρωπος που με σήκωσε από το πάτωμα όσες φορές κι αν έπεσα. Σε ευχαριστώ. Χωρίς εσένα θα είχα μείνει πίσω. Πολύ πίσω... Χρήστο, αν αληθεύει ότι το γέλιο σού δίνει χρόνια ζωής, χάρη σε εσένα θα ζήσω μέχρι τα 125. Σε ευχαριστώ, φίλε μου, για όλα, κι εύχομαι να συνεχίσουμε να είμαστε "φοιτητοσυνεργάτες" και στο μέλλον... Γιώργο, ήσουν ο πρώτος μου φίλος στην Αθήνα. Λένε ότι οι παρέες συνήθως αλλάζουν καθώς

περνάει ο πρώτος καιρός, αλλά εσύ μου έδειξες πόσο τυχερός ήμουν από την πρώτη μέρα. Κι αν χωρίζουν οι δρόμοι μας, ξέρω ότι δε θα σε χάσω. Έχουμε και plan B... Μαράκ', εμείς οι δύο μοιάζουμε και το ξέρεις. Μερικές φορές, αισθάνομαι ότι γνωριζόμαστε από πάντα. Η τάση σου να διοχετεύεις την ευφυία σου στο να βοηθάς με εντυπωσιάζει και την λατρεύω. Από τους πιο ικανούς ανθρώπους που έχω γνωρίσει, ξέρω ότι θα τα καταφέρεις. Δεν γίνεται αλλιώς...

Δημήτρη, συνονόματε φίλε κι αδερφέ μου, από το δημοτικό μέχρι σήμερα. Μπορεί να απέχουμε περίπου 500 χιλιόμετρα εδώ και πέντε χρόνια, αλλά δεν είμαστε όντως μακρυά. Είσαι ο παιδικός μου φίλος κι εύχομαι να μη σε χάσω ποτέ. Σε θαυμάζω, αδερφέ μου, και πιστεύω ότι θα φτάσεις ψηλά, πολύ ψηλά. Κάθε επιτυχία σου, τη νιώθω σα δική μου. Σε ευχαριστώ που είσαι φίλος μου...

Πείτε με μελοδραματικό, που μπορεί και να είμαι... Αλλά αγαπώ τους ανθρώπους μου, και θέλω να τους το λέω.

Αθήνα, Οκτώβριος 2022

Δημήτριος Δήμος

Contents

Περίληψη 7

Abstract 9

Ευχαριστίες 13

I Εκτεταμένη Περίληψη στα Ελληνικά 25

1 Μοντέλα score-based και diffusion 27
1.1 Περιγραφή Μοντέλων Score-based . . . 27
1.1.1 Δειγματολήπτης Annealed Langevin Dynamics . . . 28
1.2 Περιγραφή μοντέλων diffusion . . . 28
1.2.1 Denoising Diffusion Implicit Model (DDIM) . . . 31

2 Μοντέλα diffusion μέσω Στοχαστικών Διαφορικών Εξισώσεων 33
2.1 Σύνδεση μοντέλων score-based και diffusion με ΣΔΕ . . . 33

3 Προβλήματα αντιστροφής με μοντέλα diffusion 35
3.1 Compressed Sensing σε Μαγνητικές Τομογραφίες . . . 35
3.2 Compressed Sensing σε MRI με score-based μοντέλα . . . 36

4 Συνεισφορά και Πειράματα 39
4.1 Ισοδυναμία Variance Exploding και Variance Preserving μοντέλων diffusion . 39
4.2 Σχέση μεταξύ εξόδων των score-based and diffusion μοντέλων . . . 40
4.3 Μετασχηματίζοντας τον DDIM σε Score-based Implicit Model (SBIM) . . . 40
4.4 Γραμμικά προβλήματα αντιστροφής μέσω Score-based Implicit Model . . . 41
4.4.1 Καθοδηγούμενος Score-based implicit model . . . 41
4.4.2 Compressed Sensing σε MRI μέσω καθοδηγούμενου score-based implicit model . . . 42

II Theoretical Background 45

5 Introduction 47
5.1 Thesis Objective . . . 48
5.2 Thesis Organization . . . 49

6 Score-based Generative Modeling **51**
6.1 Networks that model gradients of the data distribution . . . 51
6.2 Score matching for score estimation . . . 51
6.2.1 Denoising Score Matching . . . 52
6.2.2 Sliced Score Matching . . . 52
6.3 Langevin Dynamics . . . 52
6.4 Noise Conditional Score Networks (NCSNs) . . . 53
6.5 Learning NCSNs via score matching . . . 53
6.6 NCSN inference via annealed Langevin Dynamics . . . 54

7 Diffusion Models **55**
7.1 Why diffusion? . . . 55
7.2 Forward diffusion process . . . 56
7.3 Reverse diffusion process . . . 57
7.4 Parameterization of the training objective . . . 59
7.4.1 Training loss simplification . . . 60
7.5 Denoising Diffusion Implicit Models . . . 60
7.5.1 Sampler derivation . . . 60
7.6 Diffusion Models Summary . . . 61

8 Score-based Generative Modeling with Stochastic Differential Equations **63**
8.1 Perturbing data with an SDE . . . 63
8.2 Variance Exploding and Variance Preseving SDEs . . . 65
8.3 Reversing the SDE for sample generation . . . 65
8.3.1 General purpose numerical SDE solvers . . . 66
8.3.2 Predictor-Corrector samplers . . . 66

9 Compressed Sensing **69**
9.1 Methodology . . . 69
9.1.1 Underdetermined Linear Systems . . . 69
9.1.2 General Solution Method . . . 70
9.2 Compressed Sensing MRI . . . 71
9.2.1 Problem Statement . . . 71
9.2.2 Multi-coil Magnetic Resonance Imaging . . . 73
9.3 Compressed Sensing MRI with score-based models . . . 73

III Contribution and Experiments **75**

10 Bijective Relation between score-based and diffusion models **77**
10.1 Equivalence between Variance Exploding and Variance Preserving diffusion models . . . 77
10.2 Relation between score-based and diffusion model outputs . . . 78
10.3 Transforming DDIM into Score-based Implicit Model (SBIM) . . . 78

11 Solving linear inverse problems via Score-based Implicit Model **81**
11.1 Guided score-based implicit model 81
11.2 One extra denoising step 81
11.3 Guided score-based implicit model with Langevin Dynamics corrector . . . 82

12 Compressed Sensing MRI via guided score-based implicit model **83**
12.1 NYU fastMRI Dataset 83
12.2 Model Architecture 83
12.3 Experimental Results 84
12.3.1 In-distribution (brain) - Vertical Masking 84
12.3.2 In-distribution (brain) - Horizontal Masking 86

13 Discussion **87**
13.1 Related Work 87
13.2 Conclusion 87

Appendices **89**

A Score-based Implicit Model Sample Sample Reconstructions **91**

B Guided score-based implicit model with LD corrector Sample Reconstructions **97**

C Langevin Dynamics Sample Sample Reconstructions **103**

Βιβλιογραφία **113**

List of Figures

1.1 Ευθεία και Αντίστροφη Διάχυση 29
1.2 Οπτικοποίηση διάχυσης σε 2D swiss roll δεδομένα. [1] 30

4.1 Σύγκριση Αλγορίθμων - Κατακόρυφη Μάσκα 42
4.2 Sampler Comparison - Horizontal Mask 44

6.1 Three random sampling trajectories generated with Langevin dynamics [2] 54

7.1 Forward and Reverse Diffusion Processes 56
7.2 Training a diffusion model for modeling a 2D swiss roll data. [1] 59

8.1 Overview of score-based generative modeling through SDEs [3] 64
8.2 Solving a reverse SDE yields a score-based generative model [3] 66

9.1 Underdetermined Linear System: $x = A \cdot y$ 70
9.2 Toy example of Compressed Sensing 71

12.1 Sampler Comparison - Vertical Mask 84
12.2 Sampler Comparison - Horizontal Mask 86

A.1 R=2, 32 steps (part 1) 91
A.2 R=2, 32 steps (part 2) 92
A.3 R=4, 32 steps (part 1) 93
A.4 R=4, 32 steps (part 2) 94
A.5 R=12, 32 steps (part 1) 95
A.6 R=12, 32 steps (part 2) 96

B.1 R=2, 32 predictor steps + 64 corrector steps (part 1) 97
B.2 R=2, 32 predictor steps + 64 corrector steps (part 2) 98
B.3 R=4, 32 predictor steps + 64 corrector steps (part 1) 99
B.4 R=4, 32 predictor steps + 64 corrector steps (part 2) 100
B.5 R=12, 32 predictor steps + 64 corrector steps (part 1) 101
B.6 R=12, 32 predictor steps + 64 corrector steps (part 2) 102

C.1 R=2, 268 steps (part 1) 103
C.2 R=2, 268 steps (part 1) 104
C.3 R=4, 268 steps (part 1) 105
C.4 R=4, 268 steps (part 1) 106

C.5 R=12, 268 steps (part 1) 107
C.6 R=12, 268 steps (part 1) 108

List of Tables

4.1 Σύγκριση Αλγορίθμων - SSIM - Κατακόρυφη Μάσκα 43
4.2 Σύγκριση Αλγορίθμων - PSNR - Κατακόρυφη Μάσκα 43
4.3 Sampler Comparison - Horizontal Mask 44

12.1 Sampler Comparison - SSIM - Vertical Mask 85
12.2 Sampler Comparison - PSNR - Vertical Mask 85
12.3 Sampler Comparison - Horizontal Mask 86

List of Algorithms

6.1 Annealed Langevin Dynamics 54
7.1 Training diffusion models [4] 61
7.2 Ancestral sampling [4] 61
7.3 Denoising Diffusion Implicit Model [5] 61
8.1 PC Sampling (VE SDE) 67
8.2 PC Sampling (VP SDE) 67
8.3 Corrector Algorithm (VE) 67
8.4 Corrector Algorithm (VP) 67
9.1 Guided Langevin Dynamics 74
10.1 Score-based Implicit Model 80
11.1 Guided score-based Implicit Model 82
11.2 Guided score-based implicit model with LD corrector 82

Part I

Εκτεταμένη Περίληψη στα Ελληνικά

Chapter 1

Μοντέλα score-based και diffusion

Τα μοντέλα score-based και diffusion συχνά αποκαλούνται απλά μοντέλα diffusion θεωρώντας πως το ίδιο όνομα περιγράφει και τις δύο κατηγορίες μοντέλων. Αυτό είναι λογικό, καθώς οι ποσότητες που οι δύο τύποι μοντέλων παράγουν στην έξοφδό τους αποδεθκνύεται ότι είναι ισοδύναμες. Τα μοντέλα diffusion έχουν επιτύχει σημαντική πρόοδο σε τομείς όπως η παραγωγή εικόνας, ήχου, σχήματος και μουσικής, ενώ η ποιότητα των παραγώμενων εικόνων τους έχει νικήσει μέχρι και αυτή των GANs. Αυτή η ιλιγγιώδης πρόοδος έχει, μάλιστα, επιτευχθεί εντός ολίγων χρόνων. Εν γένει, τα μοντέλα αυτά αποτελούνται από δύο κύριες συνιστώσες: ένα νευρωνικό δίκτυο κι έναν επαναληπτικό αλγόριθμο παραγωγής δειγμάτων.

1.1 Περιγραφή Μοντέλων Score-based

Θεωρούμε μια άγνωστη κατανομή δεδομένων που είναι i.i.d. $\{x_i \in \mathbb{R}\}_{i=1}^{N}$. Τα μοντέλα score-based προσπαθούν νε εκτιμήσουν μια συνάρτηση, που ονομάζεται συνάρτηση score, που αναπαριστά την κατανομή δεδομένων μέσω του $\nabla_x \log p_{data}(x)$. Δηλαδή, η συνάρτηση score ορίζεται ως: $s(x) = \nabla_x \log p(x)$. Η μέθοδος μέσω της οποίας τα μοντέλα αυτά καταφέρουν να μάθουν αυτή την ποσότητα ονομάζεται score-matching [6]. Αποδεικνύεται [6] η συνάρτηση που θεωρητικά πρέπει να ελαχιστοποιήσουμε για να μαθευτεί η score-συνάρτηση της κατανομής είναι η:

$$\mathbb{E}_{p_{data}}\left[\text{tr}(\nabla_x s_\partial(x)) + \frac{1}{2}\|s_\partial(x)\|_2^2\right]$$

η οποία, ωστόσο, δεν μπορεί να υπολογιστεί για κατανομές μεγάλης κλίμακας λόγω του υψηλού υπολογιστικού κόστους που απαιτεί ο όρος $\text{tr}(\nabla_x s_\partial(x))$. Οι πιο γνωστοί τρόποι υλοποίησης του score-matching είναι οι denoising score-matching [7] και sliced score-matching [8], με τον πιο δημοφιλή εκ των δύο να είναι ο denoising score-matching.

Για την υλοποίηση του denoising score-matching προσθέτουμε προκαθορισμένης και γνωστής κατανομής θόρυβο στα δεδομένα και παίρνουμε μια θορυβοποιημένη κατανομή δεδομένων: $q_\sigma(\tilde{x}) \triangleq \int q_\sigma(\tilde{x}|x)p_{data}(x)\,dx$. Για να μάθουμε την score συνάρτηση της θορυβοποιημένης κατανομής αυτής, χρειάζεται να ελαχιστοποιήσουμε την εξής ποσότητα:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{data}(x)}\left[\|s_\partial(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2\right]$$

Όταν η διασπορά του θορύβου πλησιάζει στο 0, τότε το μοντέλο μας προσεγγίζει την κατανομή των δεδομένων που δεν έχουν θόρυβο.

1.1.1 Δειγματολήπτης Annealed Langevin Dynamics

Αν θεωρήσουμε δεδομένο ότι έχουμε πρόσβαση στην score συνάρτηση μιας κατανομής ενδιαφέροντος, τότε μπορούμε να παράξουμε νέα δεδομένα με χρήση αυτής. Ο πρώτος τρόπος που προτάθηκε για το προαναφερθέν επιχείρημα είναι η δειγματοληψία μέσω Langevin Dynamics. Ο αλγόριθμος Langevin Dynamics είναι επαναληπτικός και προσπαθεί να λάβει δείγματα από την κατανομή της οποίας γνωρίζουμε την score συνάρτηση μέσω της σχέσης:

$$\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x log p(\tilde{x}_{t-1}) + \sqrt{\epsilon}\, z_t$$

Όπως αναφέραμε προηγουμένως, δεν μπορούμε να έχουμε αμέσως πρόσβαση στην score συνάρτηση, γι'αυτό και χρειάζεται να θορυβοποιήσουμε την κατανομή δεδομένων και να μάθουμε την score συνάρτηση του αποτελέσματος. Εκμεταλλευόμενοι αυτή την ιδιότητα και την ιδές των Langevin Dynamics, οι Song και Ermon [9] πρότειναν τα μοντέλα Noise Conditional Score Networks (NCSNs).

Θεωρούμε μια διακριτή ακολουθία από τυπικές αποκλίσεις (κλίμακες) $\sigma_1, \sigma_2, \ldots, \sigma_N$. Τα μοντέλα NCSN προσπαθούν να εκτιμήσουν την score συνάρτηση από κάθε κατανομή που προκύπτει αν θορυβοποιήσουμε την αρχικη κατανομή δεδομένων με θόρυβο Gauss τυπικής απόκλισης που ανήκει στην παραπάνω προκαθορισμένη ακολουθία. Για να το πετύχουν αυτό, χρησιμοποιούν τη μέθοδο denoising score-matching (ή την sliced score-matching) για κάθε κλίμακα θορύβου. Η συνάρτηση ελαχιστοποίησης είναι η:

$$\ell(\theta, \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{data}}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2 I)}\left[\left\| s_\theta(\tilde{x}, \sigma) + \frac{\tilde{x} - x}{\sigma^2}\right\|^2\right]$$

και μετά την ενοποίηση των συναρτήσεων για κάθε κλίμακα, έχουμε:

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^L) \triangleq \frac{1}{L}\sum_{i=1}^{L} \lambda(\sigma_i)\ell(\theta, \sigma_i)$$

Επομένως, τα μοντέλα NCSN μπορούν να εκτιμούν στην έξοδό τους την ποσότητα: $s_\theta(x) \approx \nabla_x \log p(x)$. Επομένως, μπορούμε να ενσωματώσουμε τα μοντέλα στον αλγόριθμο Langevin Dynamics για να παράγουμε δεδομένα απο την κατανομής μας. Ξεκινάμε από τυχαίο Gauss θόρυβο και σε κάθε βήμα του Langevin Dynamics μειώνουμε την κλίμακα θορύβου σ κατά ένα επίπεδο. Το σημαντικό σημείο που πρέπει να προσέξουμε είναι ότι σε κάθε βήμα του Langevin Dynamics (που αντιστοιχεί σε μια κλίμακα θορύβου) μπορεί να επιλέξουμε να κάνουμε κάποιες υποεπαναλήψεις, για να πετύχουμε καλύτερη σύγκλιση σε δεδομένο εντός κατανομής. Ο αλγοριθμος παρουσιάζεται και συμπαγώς στο 6.1.

1.2 Περιγραφή μοντέλων diffusion

Τα μοντέλα diffusion [4] ακολουθούν διαφορετική λογική από τα score-based για να παράξουν δεδομένα. Όπως προηγουμένως, θεωρούμε μια κατανομή i.i.d. δεδομένων και

μια ακολουθία διακριτών τυπικών αποκλίσεων (κλιμάκων) θορύβου: $\beta_1, \beta_2, \ldots, \beta_T$. Αυτή η ακολουθία είναι χαρακτηριστική των diffusion μοντέλων που θα περιγραψουμε στη συνέχεια.

Τα μοντέλα diffusion προσπαθούν να μιμηθούν την αντίστροφη διαδικασία του φαινομένου της διάχυσης. Ας θεωρήσουμε την ευθεία διάχυση: προσθέτουμε όλο και περισσότερο θόρυβο στα αρχικά καθαρά δεδομένα, ο οποίος θόρυβος έχει σε κάθε βήμα τυπική απόκλιση β_t. Η κατανομή από την οποία λαμβάνουμε δείγμα σε κάθε επανάληψη είναι η:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

και η απο κοινού κατανομή ολόκληρης της διαδικασίας, που ονομάζεται και *ευθεία διάχυση* είναι η:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

Καθώς το t μεγαλώνει, τα δεδομένα γίνονται όλο και πιο θορυβώδη. Αν θεωρήσουμε $T \to \infty$ τότε στο τέλος τα δεδομένα θα είναι σαν ληφθέντα από μια ισοτροπική γκαουσιαννή κατανομή. Τα μοντέλα diffusion προσπαθούν να εκτιμήσουν τις επιμέρους κατανομές από τις οποίες πρέπει να λάβουμε δείγματα ξεκινώντας από γκαουσιανό θόρυβο για να φτάσουμε πίσω στα καθαρά, δηλαδή την αντίστροφη διάχυση. Η ιδέα αυτή απεικονίζεται στο ακόλουθο σχήμα:

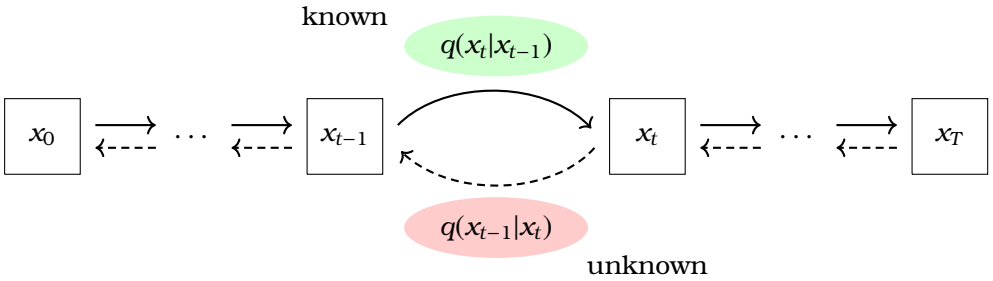


Figure 1.1: *Ευθεία και Αντίστροφη Διάχυση*

Γενικώς, σύμφωνα με το Λήμμα 7.1 και την Απόδειξη 7.2, έχουμε τη δυνατότητα να εκτιμήσουμε σε ένα βήμα οποιοδήποτε στάδιο της εξέλιξης ενός δεδομένου που υφίσταται την ευθεί διάχυση. Η σχέση που μας δίνει το ζητούμενο προκύπτει να είναι η:

$$x_t = \sqrt{\bar{a}_t}\, x_0 + \sqrt{1-\bar{a}_t}\, \epsilon$$

όπου $\bar{a}_t = \prod_{i=1}^{t}(1-\beta_i)$.

Μοντελοποιούμε τη αντίστροφη διάχυση ως μια διαδιχή κατανομών:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Η αντίστροφη υπό συνθήκη κατανομή είναι ανιχνεύσιμη όταν είναι γνωστό το x_0:

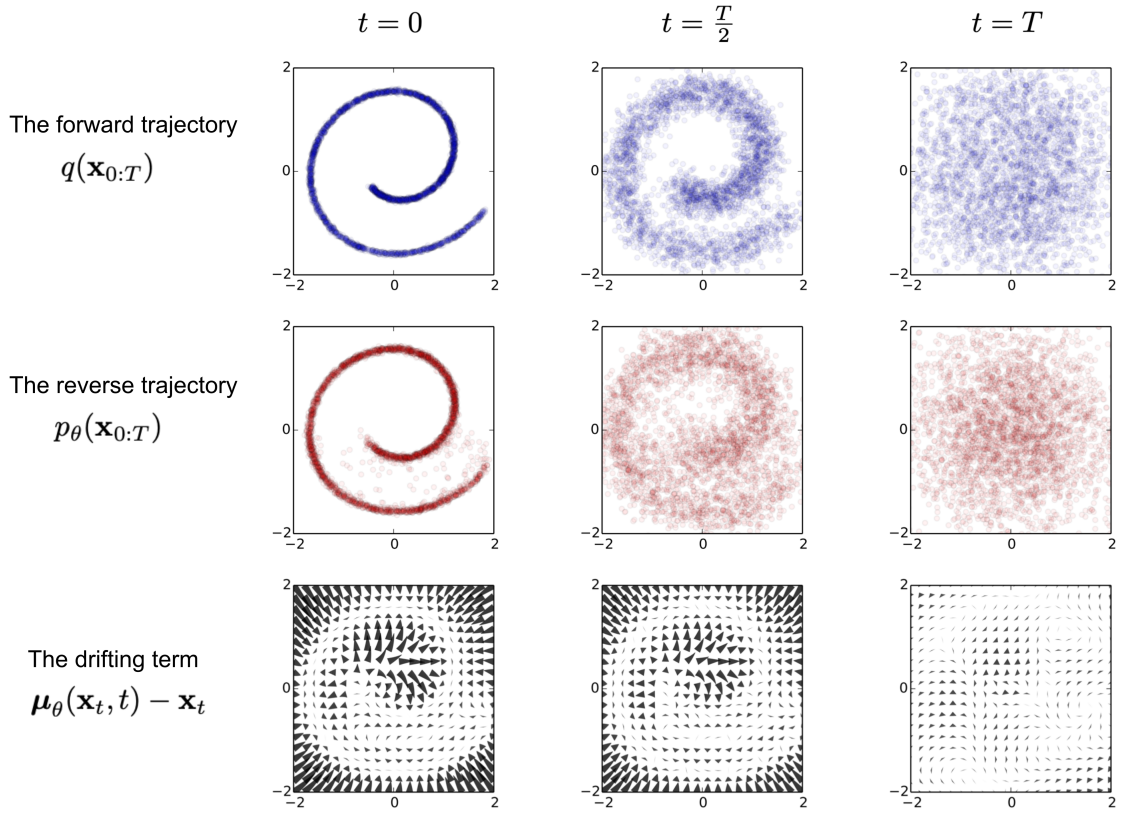


Figure 1.2: *Οπτικοποίηση διάχυσης σε 2D swiss roll δεδομένα. [1]*

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I})$$

Με χρήση αυτής της ιδιότητας και τον κανόνα του Bayes, μπορούμε να μοντελοποιήσουμε τις μέσες τιμές και διασπορές των ενδιάμεσων βημάτων της αντίστροφης διάχυσης ως:

$$\tilde{\beta}_t = \frac{1 - \bar{a}_{t-1}}{1 - \bar{a}_t} \cdot \beta_t$$
$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{a_t}(1 - \bar{a}_{t-1})}{1 - \bar{a}_t} x_t + \frac{\sqrt{\bar{a}_{t-1}}\beta_t}{1 - \bar{a}_t} x_0$$

Με χρήση του Λήμματος 7.1, μπορούμε να εκφράσουμε το x_0 στη μορφή:

$$x_0 = \frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1 - \bar{a}_t} z_t)$$

Εισάγοντας την παραπάνω έκφραση στην μέση τιμή, παίρνουμε:

$$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1 - a_t}{\sqrt{1 - \bar{a}_t}} z_t\right)$$

Εφόσον έχουμε πρόσβαση στο x_t, θέλουμε να ξέρουμε την ποσότητα θορύβου z_t για να προσδιορίσουμε τη μέση τιμή μ_θ. Εκπαιδεύουμε λοιπόν το diffusion μοντέλο να προβλέπει το z_t δοθέντος του x_t. Έχουμε, λοιπόν:

$$\mu_\partial(x_t, t) = \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_\partial(x_t, t)\Big)$$

Άρα:

$$x_{t-1} = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_\partial(x_t, t)\Big), \Sigma_\partial(x_t, t))$$

Η τελική συνάρτηση βελτιστοποίησης (ύστερα από απλοποιήσεις που οι Ho et al. [4] παρατήρησαν ότι οδηγούν σε αποτελεσματικότερη εκπαίδευση) αποσκοπεί στην ελαχιστοποίηση της διαφοράς προστιθέμενου θορύβου από τον θόρυβο που το μοντέλο προβλέπει ότι έχει προστεθεί:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],x_0,z_t}\Big[\|z_t - z_\partial(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}z_t, t)\|^2\Big] + C$$

όπου C μια σταθερά ανεξάρτητη του ∂.

Η διασπορές των αντίστροφων κατανομών διάχυσης μπορεί να αναπαρασταθεί με ποικίλους τρόπους. Ένας συνήθης είναι να θεωρηθεί ως μια προκαθορισμένη διακριτή γραμμικά αυξανόμενη ακολουθία T τυπικών αποκλίσεων, με πιο δημοφιλείς οριακές τιμές $\beta_1 = 10^{-4}$ και $\beta_T = 0.02$.

Ο πρώτος αλγόριθμος δειγματοληψίας με χρήση μοντέλων diffusion (που ουσιαστικά πραγματοποιεί ακριβώς την ίδια διαδικασία με την ευθεία διάχυση, αλλά σε πολλά βήματα και με βάση την προβλεπόμενη μέση τιμή και την προκαθόρισμένη διασπορά θορύβου) είναι ο Ancestral Sampling και παρουσιάζεται στο 7.2.

1.2.1 Denoising Diffusion Implicit Model (DDIM)

Ο Denoising Diffusion Implicit Model (DDIM) είναι ένας μεταγενέστερος αλγόριθμος δειγματοληψίας για μοντέλα diffusion που, όμως, είναι εγγενώς πιο γρήγορος. Αυτό σημαίνει ότι παράγει πολύ καλά αποτελέσματα εάν εφαρμοστεί για ένα υποσύνολο των διαδοχικών κατανομών της αντίστροφης διάχυσης, παραλείποντας πολλά ενδιάμεσα στάδια. Μάλιστα, ο εν λόγω αλγόριθμος τις περισσότερες φορές αποτυγχάνει όταν εκτελείται για πολλά βήματα. Ο αλγόριθμος παράγεται σύμφωνα με το παρακάτω σκεπτικό:

Επαναπαραμετροποιούμε τις κατανομές αντίστροφης διάχυσης q_σ, σύμφωνα με το Λήμμα 7.1:

$$\begin{aligned}
x_{t-1} &= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}}z_{t-1} \\
&= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}z_t + \sigma_t z \\
&= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}\frac{x_t - \sqrt{\bar{a}_t}x_0}{\sqrt{1-\bar{a}_t}} + \sigma_t z \\
q_\sigma(x_{t-1}|x_t, x_0) &= \mathcal{N}(x_{t-1}; \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}\frac{x_t - \sqrt{\bar{a}_t}x_0}{\sqrt{1-\bar{a}_t}}, \sigma_t^2\mathbf{I})
\end{aligned}$$

Ισχύει ότι $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I})$, άρα:

$$\tilde{\beta}_t = \sigma_t^2 = \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\cdot\beta_t$$

Έστω $\sigma_t^2 = \eta \cdot \tilde{\beta}_t$ ώστε να μπορούμε να επιλέξουμε $\eta \in \mathbb{R}^+$ ως υπερπαράμετρο για να ελέγξουμε την στοχαστικότητα της δειγματοληψίας. Στην ειδική περίπτωση του $\eta = 0$ η διαδικασία δειγματοληψίας γίνεται ντετερμινιστική και το μοντέλο διάχυσης γίνεται denoising diffusion implicit model [5].

Κατά τη παραγωγή, χρησιμοποιούμε ένα υποσύνολο S βημάτων διάχυσης $\{\tau_1, \ldots, \tau_S\}$ και η διαδικασία παραγωγής γίνεται:

$$q_{\sigma,\tau}(x_{\tau_{i-1}}|x_{\tau_t}, x_0) = \mathcal{N}(x_{\tau_{i-1}};\ \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1 - \bar{a}_{t-1} - \sigma_t^2}\frac{x_{\tau_i} - \sqrt{\bar{a}_t}x_0}{\sqrt{1 - \bar{a}_t}}, \sigma_t^2\mathbf{I})$$

Έχει παρατηρηθεί εμπειρικά ότι ο DDIM παράγει καλύτερα αποτελέσματα όταν εκτελείται σε ένα υποσύνολο των T αρχικών βημάτων διάχυσης στα οποία έχει εκπαιδευτεί αρχικά το μοντέλο, ενώ ο Ancestral Sampling παράγει καλύτερα αποτελέσματα όταν εκτελείται σε όλα τα T βήματα. Με τον DDIM, μπορούμε να εκπαιδεύσουμε ένα diffusion μοντέλο σε έναν αυθαίρετο αριθμό βημάτων, αλλά να παράξουμε δεδομένα χρησιμοποιώντας ένα υποσύνολό τους.

Chapter 2

Μοντέλα diffusion μέσω Στοχαστικών Διαφορικών Εξισώσεων

2.1 Σύνδεση μοντέλων score-based και diffusion με ΣΔΕ

Τα μοντέλα score-based και diffusion παρουσιάζουν πολλές ομοιότητες ως προς τη διαδικασία παραγωγής δεδομένων. Τα score-based μοντέλα παράγουν δεδομένα μέσω Langevin Dynamics "πηγαίνοντας" από θορυβοποιημένες κατανομές σε κατανομές με όλο και λιγότερο θόρυβο διασχίζοντας ουσιαστικά τη θορυβοποιημένη συνάρτηση score. Τα diffusion μοντέλα ξεκινούν από γκαουσιανό θόρυβο και βήμα-βήμα αποθορυβοποιούν τα παραγώμενα δεδομένα μέχρι να "καθαρίσουν" εντελώς. Ο λόγος που αυτές οι διαδικασίες μοιάζουν φανερώθηκε στο έργο των Song et al. [3], όπου και δείχτηκε ότι οι διαδικασίες παραγωγής των μοντέλων αυτών είναι στην πραγματικότητα επίλυση συγκεκριμένων Στοχαστικών Διαφορικών Εξισώσεων (ΣΔΕ).

Η γενική μορφή μιας ΣΔΕ είναι:

$$dx = f(x,t)dt + g(t)dw$$

όπου $f(\cdot,t) : \mathbb{R}^d \to \mathbb{R}^d$ είναι μια διανυσματική συνάρτηση με όνομα συντελεστής drift, $g(t) \in \mathbb{R}$ μια πραγματική συνάρτηση με όνομα συντελεστής διάχυσης, w είναι η κλασσική κίνηση Brown [10] και dw είναι μια απειροστή ποσότητα λευκού θορύβου. Η λύση της ΣΔΕ είναι μια συνεχής συλλογή τυχαίων μεταβλητών $\{x(t)\}_{t\in[0,T]}$ οι οποίες ακολουθούν στοχαστικές τροχιές καθώς ο χρόνος πηγαίνει από το 0 στο T. Η εικόνα 8.1 οπτικοποιεί τη γενική αυτή περιγραφή.

Για τα μοντέλα score-based, γνωρίζουμε ότι η κατανομή του x_i ($i \in [1,N]$) αντιστοιχεί στην αλυσίδα Markov:

$$x_t = x_{t-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}$$

όπου $z_i \sim \mathcal{N}(0,I)$ και έχουμε εισάγει το $\sigma_0 = 0$ για απλοποίηση της ορολογίας. Στο όριο $N \to \infty$, οι διασπορές σ_i^2, ο θόρυβος z_i και η αλυσίδα $\{x_i\}_{i=0}^T$ γίνονται συνεχείς συναρτήσεις του χρόνου $t \in [0,T]$ και η διαδικασία $\{x(t)\}_{t=0}^1$ είναι η λύση της παρακάτω ΣΔΕ:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw$$

Εντελώς όμοια, η ευθεία διάχυση αντιστοιχεί στην παρακάτω αλυσίδα Markov:

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i}z_{i-1}, \quad i = 1, \dots, N$$

και η διαδικασία $\{x(t)\}_{t=0}^{1}$ είναι η λύση της παρακάτω ΣΔΕ:

$$dx = -\frac{1}{2}\beta(t)x\,dt + \sqrt{\beta(t)}\,dw$$

Οι διαδικασίες θορυβοποίησης των score-based και diffusion μοντέλων είναι διακριτοποιήσεις των αντίστοιχων εξισώσεων. Η ΣΔΕ των score-based μοντέλων δίνει πάντοτε μια ακολουθία από "εκρηγνυόμενες" διασπορές καθώς $t \to \infty$, ενώ η ΣΔΕ των diffusion δίνει μια διαδικασία με σταθερή διασπορά όταν η αρχική κατανομή έχει μοναδιαία διασπορά [3]. Για αυτό η πρώτη ΣΔΕ ονομάζεται Variance Exploding και η δεύτερη Variance Preserving.

Για να παράξουνε δείγματα, τα μοντέλα ουσιαστικά επιλύουν τις αντίστροφες ΣΔΕ εξ αυτών που τους αντιστοιχούν. Κάθε ΣΔΕ έχει μια αντίστροφή ΣΔΕ, η οποία σε κλειστή μορφή εκφράζεται ως:

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dw$$

Οι δειγματολήπτες των μοντέλων (Langevin Dynamics, Ancestral Sampling και DDIM) είναι ουσιαστικά ολοκληρωτές των αντίστροφων ΣΔΕ. Εφόσον γνωρίζουμε, ωστόσο, τη μορφή των ΣΔΕ, μπορούμε πλέον να χρησιμοποιήσουμε οποιοδήποτε ολοκληρωτή ΣΔΕ για να παράξουμε δείγματα. Αυτός μπορεί να είναι είτε κάποια κλασσική μέθοδος, όπως η Euler-Maruyama, είτε κάποια μέθοδος Predictor-Corrector (PC) [3], δηλαδή μια επαναληπτική διαδικασία κατά την οποία σε κάθε βήμα γίνεται μια πρόβλεψη και μερικά βήματα διόρθωσης με χρήση Langevin Dynamics. Οι δειγματολήπτες PC που προτείνονται από τους Song et al. [3] παρουσιάζονται στο τέλος του κεφαλαίου 8.

Chapter 3

Προβλήματα αντιστροφής με μοντέλα diffusion

Η οικογένεια των μοντέλων diffusion είναι πολλά υποσχόμενη για την παραγωγή εικόνας και έχει ήδη φανεί ιδιαίτερα χρήσιμη και σε άλλους τομείς, όπως είναι τα προβήματα αντιστροφής. Ως πρόβλημα αντιστροφής, εν γένει, ορίζεται το πρόβλημα εντοπισμού της αιτίας δοθείσας της παρατήρησης ενός γεγονότος. Ένα τέτοιο πρόβλημα είναι η επίλυσης ενός γραμμικού συστήματος που έχει περισσότερους αγνώστους από εξισώσεις, το οποίο έχει γενικά άπειρες λύσεις. Μια τεχνική για να ποσεγγίσουμε τέτοια προβήματα είναι το Compressed Sensing, η οποία επιτρέπει να βρούμε μια λύση η οποία είναι "αραιή" (sparse), δηλαδή έχει σημαντικά περισσότερα μηδενικά στοιχεία από μη μηδενικά. Υπάρχουν διάφορες υλοποιήσεις του compressed sensing [11], ωστόσο σε αυτό το κεφάλαιο θα εστιάσουμε σε προσεγγίσεις προβλημάτων αντιστροφής με μοντέλα diffusion.

3.1 Compressed Sensing σε Μαγνητικές Τομογραφίες

Αρχικά ορίζουμε το compressed sensing σε μαγνητικές τομογραφίες (MRIs). Στόχος των MRI είναι να παράγουν μια εικόνα της πυκνότητας των ιστών στο σώμα ενός ασθενή. Ορίζουμε μια εικόνα ως $x \in C^n$: ένα n-διάστατο μιγαδικό διάνυσμα. n είναι ο αριθμός των voxel στις 3-Δ εικόνες μας. Με χρήση του MRI scanner θέλουμε να λάβουμε m δείγματα στο πεδίο της συχνότητας (k-space, or Fourier domain), με $m \ll n$ για να επιταχύνουμε το σκανάρισμα. Αναπαριστούμε τη διαδικασία λήψης μετρήσεων ως πολλαπλασιασμό του x με έναν πίνακα Φ, που κατασκευάζεται επιλέγοντας ένα υποσύνολο γραμμών του μητρικού διακριτού μετασχηματισμού Fourier (DFT). Οι μετρήσεις-δείγματα αναπαρίστανται, τώρα, ως: $y = \Phi x$. Επειδή $m \ll n$ το πρόβλημα έχει περισσότερους αγνώστους από εξισώσεις. Επειδή μας ενδιαφέρει η λύση που είναι κατά μέγιστα αραιή, η επίλυση του συστήματος για την εύρεση του x μοντελοποιείται ως το ακόλουθο πρόβλημα βελτιστοποίησης:

$$\text{Ελαχιστοποίησε } \|Wx\|_1$$

$$\text{Υπό τον περιορισμό } y = \Phi x$$

όπου η L_1 νόρμα ορίζεται ως το άθροισμα των απολύτων τιμών των στοιχείων του διανύσματος x: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$, και Wx είναι η αναπαράσταση του x σε κάποια βάση Wavelet.

Για MRI πολλαπλών πηνίων, το πρόβλημα επαναδιατυπώνεται ως εξής [12]: οι μετρήσεις

$y_i \in C^L$ που λαμβάνονται από το i-οστό πηνίο περιγράφονται από το ευθύ μοντέλο [13]:

$$y_i = PFS_i x^* + w_i, \ i = 1, \dots, N_c \tag{3.1}$$

όπου $x^* \in C^N$ είναι η N-pixel εικόνα, S_i είναι τελεστής που αναπαριστά σημείο-προς-σημείο πολλαπλασιασμό του χάρτη ευσιασθησίας του i-οστού πηνίου, F είναι ο χωρικός τελεστής του M/Σ Fourier, το P αναπαριστά τον k-space τελεστή δειγματοληψίας, και θεωρούμε $w_i \sim \mathcal{N}_c(0, \sigma^2 I)$ για απλότητα. Ο ίδιος τελεστής υποδειγματοληψίας εφαρμόζεται σε όλα τα N_c πηνία.

Ο παράγοντας επιτάχυνσης R αφορά τον βαθμό υποδειγματοληψίας στον χώρο της συχνότητας. Δοθέντων μετρήσεων πολλαπλών πηνίων y, χαρτών ευαισθησίας και τελεστη δειγματοληψίας, στόχος της ανακατασκευής MRI είναι να εκτιμήσει την εικόνα x^*. Πρότερα έργα μοντελοποιούν το εν λόγω πρόβλημα ως:

$$\arg\min \|y - Ax\|^2 + \lambda Q(x), \tag{3.2}$$

όπου χρησιμοποιούμε τον τελεστή $A \in C^{M \times N}$ (with $M = N_c \cdot L$) ως γενική έκφραση όλων των διακριτά προσεγγιζόμενων γραμμικών επιδράσεων, και Q είναι ένα κατάλληλα επιλεγμένο συναρτησιακό για την εικόνα x. Για παράδειγμα, για να επιβάλουμε "αραιότητα" (sparsity), μπορούμε να "τιμωρήσουμε" την νόρμα L_1 στην wavelet αναπαράσταση του x, όπως στις Εξ. 9.1 και 9.2.

3.2 Compressed Sensing σε MRI με score-based μοντέλα

Οι Jalal et al. [12] πρότειναν μια προσέγγιση για τη λύση γραμμικών προβλημάτων αντιστροφής αξιοποιώντας τα score-based μοντέλα. Συγκεκριμένα, πρότειναν μια ενιαία μεθοδολογία για επίλυση τέτοιων προβλημάτων με εκ των υστέρων δειγματοληψία μέσω Langevin Dynamics, το οποίο μπορεί να εφαρμοστεί για την υλοποίηση Compressed Sensing ως ειδική περίπτωση.

Δοθείσας μέτρησης της μορφής:

$$y = Ax^* + w, \text{ όπου } y \in C^{M \times N}, w \sim N_c(0, \sigma^2 I) \text{ και } x^* \sim \mu \tag{3.3}$$

η εκ των υστέρων δειγματοληψία παράγει το x^* σύμφωνα με την εκ των υστέρων κατανομή $\mu(\cdot|y)$.

Σύμφωνα με την Εξ. 3.3 βλέπουμε ότι η κατανομή του y δοθέντος του x είναι αυτή της θορυβώδους συνιστώσας του, δηλαδή Γκαουσιανή. Αρα

$$\nabla_{x_t} \log p(y|x^*) = \frac{A^H(y - Ax_t)}{\sigma^2} \tag{3.4}$$

Οι Ajil et al. [12] παρατήρησαν ότι καλύτερα αποτελέσματα παράγονται όταν προστίθεται ο φθίνων με τον χρόνο όρος γ_t στον παρονομαστή: $\frac{A^H(y - Ax_t)}{\sigma^2 + \gamma_t^2}$. Από τον κανόνα του Bayes έχουμε:

$$\nabla_{x_t} log\, p(x_t|y) = \nabla_{x_t} log\, p(x_t) + \frac{A^H(y - Ax_t)}{\sigma^2 + \gamma_t^2} \tag{3.5}$$

Το παραπάνω σκεπτικό επεκτείνει τον αλγόριθμο Langevin Dynamics ενσωματώνοντας τις μετρήσεις και "καθοδηγώντας" τον ώστε να παράξει εικόνα που συμφωνεί με τις μετρήσεις y.

Chapter 4

Συνεισφορά και Πειράματα

4.1 Ισοδυναμία Variance Exploding και Variance Preserving μοντέλων diffusion

Η ακόλουθη απόδειξη είναι εμπνευσμένη από το [14] Παράρτημα Β.

Proposition 4.1. *Οι Variance Exploding και Variance Preserving παραμετροποιήσεις είναι ισοδύναμες μέσω πολλαπλασιασμού των ενδιάμεσων μεταβλητών* x_t.

Proof. Για την Variance Exploding ΣΔΕ, οι κλίμακες θορύβου είναι: $\sigma_t \in [0, +\infty)$ και για την Variance Preserving ΣΔΕ: $a_t \in (0, 1]$. Οι αντίστοιχες ενδιάμεσες κατανομές κάθε διαδικασίας είναι: $q(x_t^{VE}|x_0) = \mathcal{N}(x_t^{VE}; x_0, \sigma_t^2 I)$ και $q(x_t^{VP}|x_0) = \mathcal{N}(x_t^{VE}; \sqrt{a_t}x_0, (1 - a_t)I)$. Με χρήση του Λήμματος 7.1 εκράζουμε κάθε θορυβοποιημένο δεδομένο τη στιγμή t ως:

$$x_t^{VE} = x_0 + \sigma_t z, \quad z \sim \mathcal{N}(0, I) \tag{4.1}$$

$$x_t^{VP} = \sqrt{a_t}\, x_0 + \sqrt{1 - a_t}\, z, \quad z \sim \mathcal{N}(0, I) \tag{4.2}$$

Διαιρούμε κάθε μέλος της Εξ. 10.2 με $\sqrt{a_t}$:

$$\frac{x_t^{VP}}{\sqrt{a_t}} = x_0 + \frac{\sqrt{1 - a_t}}{\sqrt{a_t}} z = x_0 + \sqrt{\frac{1}{a_t} - 1}\, z$$

Παρατηρούμε ότι $\sqrt{\frac{1}{a_t} - 1}$ παίρνει τιμές από 0 (συμπεριλαμβανομένου) μέχρι $+\infty$ (exclusive) αντιστροφά από το σ_t.

Προφανώς, αν $\sqrt{\frac{1}{a_t} - 1} = \sigma_t$, τότε οι ποσότητες x_t^{VE} και $\frac{x_T^{VP}}{\sqrt{a_t}}$ είναι ισοδύναμες. Συνεπώς, υπάρχουν οι ακόλουθες σχέσεις μεταξύ των μοντέλων που επιλύουν τις VP και VE ΣΔΕ:

$$x_t^{VE} \longleftrightarrow \frac{x_t^{VP}}{\sqrt{a_t}} \tag{4.3}$$

$$\sigma_t \longleftrightarrow \sqrt{\frac{1}{a_t} - 1} \tag{4.4}$$

$$\frac{1}{1 + \sigma_t^2} \longleftrightarrow a_t \tag{4.5}$$

□

4.2 Σχέση μεταξύ εξόδων των score-based and diffusion μοντέλων

Ένα score-based μοντέλο εκπαιδεύεται για να εκτιμά την score συνάρτηση $s : \mathbb{R}^d \to \mathbb{R}^d$ της εισόδου x. Από την άλλη, ένα diffusion μοντέλο εκπαιδεύεται για να προβλέπει τον θόρυβο $\epsilon \sim \mathcal{N}(0, I)$ της εισόδου x. Ανάμεσα σε αυτές τις δύο εξόδους, υπάρχει μια υπονοούμενη ισοδυναμία:

Θυμηθείτε ότι $s_\theta(x) \approx \nabla_x \log p(x)$ και η κατανομή $p(x)$ είναι Γκαουσιανή. Άρα:

$$s_\theta(x) \approx \nabla_x \log p(x) = \nabla_x\Big(-\frac{1}{2\sigma^2}(x-\mu)^2\Big) = -\frac{x-\mu}{\sigma^2} = -\frac{\epsilon}{\sigma}$$

Παρατηρούμε ότι η έξοδος ενός score-based μοντέλου είναι επίσης ο θόρυβος $\epsilon \sim \mathcal{N}(0, I)$ που βρίσκεται στην είσοδό του x διαιρεμένος με την τυπική του απόκλιση.

Όμοια, θεωρώντας θορυβοποιήσεις της VP παραμετροποίησης, δηλαδή πυρήνες θορύβου $q(x_t|x_0) \sim \mathcal{N}(\sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)I)$, έχουμε:

$$s_\theta(x_t, t) \approx \nabla_{x_t} \log q(x_t) = \mathbb{E}_{q(x_0)}[\nabla_{x_t} q(x_t|x_0)] = \mathbb{E}_{q(x_0)}\Big[-\frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{a}_t}}\Big] = -\frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{a}_t}} \tag{4.6}$$

4.3 Μετασχηματίζοντας τον DDIM σε Score-based Implicit Model (SBIM)

Σε αυτή την ενότητα, μετασχηματίζουμε τον DDIM σε score-based DDIM, δηλαδή τον επαναπαραμετροποιούμε για να χρησιμοποιεί score-based μοντέλα, αλλά για να λύσει ΣΔΕ τύπου Variance Preserving

Θυμηθείτε την κύρια επαναληπτική διαδικασία του DDIM:

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, x_{0t}^{VP} + \sqrt{1-a_{t-1}}\, \epsilon_\theta(x_t^{VP}) \overset{\text{Lemma } 7.1}{\iff}$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} - \sqrt{1-a_t}\, \epsilon_\theta(x_t^{VP})}{\sqrt{a_t}} + \sqrt{1-a_{t-1}}\, \epsilon_\theta(x_t^{VP}) \overset{\text{Eq. } 10.6}{\iff}$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} + \sqrt{1-a_t} \cdot \sqrt{1-a_t}\, s_\theta(x_t^{VE}, t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}\, s_\theta(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} + (1-a_t)\, s_\theta(x_t^{VE}, t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}\, s_\theta(x_t^{VE}, t)$$

Με δεδομένο ότι το score-based μοντέλο μας είναι εκπαιδευμένο σε VE παραμετροποίηση, η είσοδός του πρέπει να είναι αντίστοιχα θορυβοποιημένη. Συναπώς, κάθε φορά που εισάγουμε το x_t στο μοντέλο, πρέπει να διαιρέσουμε με $\sqrt{a_t}$ όπως στην Eq. 4.3. Διαιρούμε κάθε μέλος του τελευταίου επαναληπτικού κανόνα με a_{t-1} **χωρίς να βλάπτεται η Variance Preserving ιδιότητα**. Ταυτόχρονα, ξέρουμε ότι x^{VE} και x^{VP} συγκλίνοουν στον ίδιο x_0, αφού το

a_t φθίνει αργά. Βάσει των παραπάνω και της Εξ. 4.3, λαμβάνουμε:

$$x_{t-1}^{VE} \leftarrow \frac{\sqrt{a_t}\, x_t^{VE} + (1-a_t)\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\, s_\partial(x_t^{VE}, t) \Longleftrightarrow$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \frac{(1-a_t)\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\, s_\partial(x_t^{VE}, t) \Longleftrightarrow$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{1-a_t}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\right] s_\partial(x_t^{VE}, t)$$

Από Εξ. 4.5, έχουμε:

$$1 - a_t = 1 - \frac{1}{1+\sigma_t^2} = \frac{\sigma_t^2}{1+\sigma_t^2} \tag{4.7}$$

Από Εξ. 4.7 και 4.5, λαμβάνουμε:

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{\frac{\sigma_t^2}{1+\sigma_t^2}}{\sqrt{\frac{1}{1+\sigma_t^2}}} - \frac{\sqrt{\frac{\sigma_{t-1}^2}{1+\sigma_{t-1}^2}} \cdot \sqrt{\frac{\sigma_t^2}{1+\sigma_t^2}}}{\sqrt{\frac{1}{1+\sigma_{t-1}^2}}}\right] s_\partial(x_t^{VE}, t) \Longleftrightarrow$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{\sigma_t^2}{\sqrt{1+\sigma_t^2}} - \frac{\sigma_{t-1} \cdot \sigma_t^2}{\sqrt{1+\sigma_t^2}}\right] s_\partial(x_t^{VE}, t) \Longleftrightarrow$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\Big[\sigma_t - \sigma_{t-1}\Big]\, s_\partial(x_t^{VE}, t)$$

Συναπώς, μπορούμε να χρησιμοποιήσουμε ένα score-based μοντέλο για να υλοποιήσουμε δειγματοληψία με DDIM, ο οποίος έχει VP παραμετροποίηση, σύμφωνα με τον αλγόριθμο 10.1.

4.4 Γραμμικά προβλήματα αντιστροφής μέσω Score-based Implicit Model

4.4.1 Καθοδηγούμενος Score-based implicit model

Εμπνεόμενοι από το Ajil et al. [12], επεκτείνουμε τον Αλγόριθμο 10.1 ώστε να ενσωματώνει μετρήσεις στην επαναληπτική διαδικασία. Τα ίδια μαθηματικά περιγράφουν και αυτή την περίπτωση, όπως περιγράφηκαν προηγουμένως. Προκειμένου να ενσωματώσουμε τις μετρήσεις, θέλουμε να λάβουμε δείγματα από την $p(x|y)$, η οποία αντιστοιχεί στην score-συνάρτηση της Εξ. 9.7. Θυμηθείτε ότι:

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \frac{A^H(y - Ax)}{\sigma^2 + \gamma_t^2}$$

Βάσει του ίδιου σκεπτικού, προσθέτουμε την ποσότητα καθοδήγησης στον αλγόριθμο SBIM για να υλοποιήσουμε τον *Καθοδηγούμενο score-based implicit model*. Ο αλγόριθμος δίνεται ως Αλγόριθμος 11.1, με μια προσθήκη στο τελευταίο βήμα: επιστρέφουμε την ποσότητα $x_0 + \sigma_0^2 s_\theta(x_0, \sigma_0)$, που ουσιαστικά αφαιρεί τον θόρυβο $\mathcal{N}(0, \sigma_0^2 I)$ from x_0 βάσει του τυπου Tweedie [15].

Εμπνεόμενοι από τους αλγορίθμους Predictor-Corrector των Song et al. [3], επεκτείνουμε τον Αλγόριθμο Καθοδηγούμενου SBIM με την προσθήκη ενός Langevin Dynamics Corrector μετά από κάθε επανάληψη. Η εν λόγω επέκταση παρουσιάζεται στο 11.2 και δεν είναι πλέον ντετερμινιστικός.

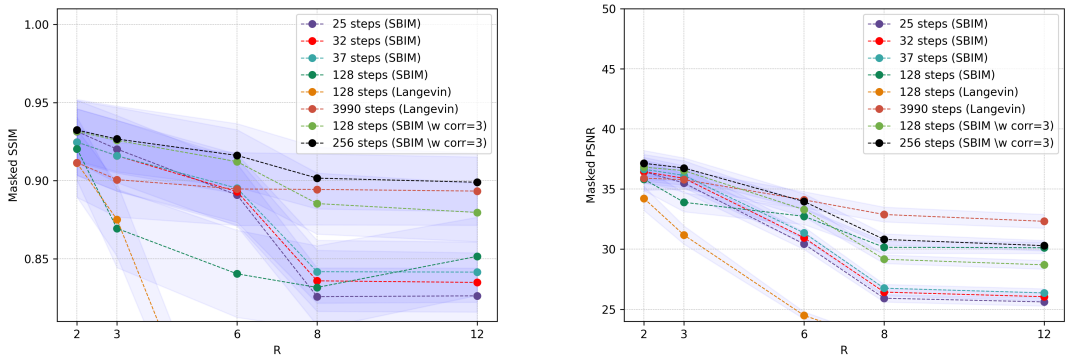
4.4.2 Compressed Sensing σε MRI μέσω καθοδηγούμενου score-based implicit model

Εντός κατανομής (εγκάφαλοι) - Κατακόρυφες Μάσκες

Για τα πειράματά μας χρησιμοποιούμε το dataset NYU fastMRI που περιέχει MRI εγκεφάλων. Το μοντέλο score-based που χρησιμοποιούμε είναι προεκπαιδευμένο σε MVUE εγκεφάλων και δημόσια προσβάσιμο στο επίσημο repository κώδικα του [12]. Ο κώδικας για τα πειράματά μας βρίσκεται στο repository: `https://github.com/d-dimos/thesis_ntua_sbim`.

Αξιολογούμε τους δειγματολήπτες Langevin Dynamics, SBIM και SBIM με Langevin Dynamics Corrector αναφορικά με την επίδοσή τους στο να υλοποιήσουν Compressed Sensing βάσει των μετρικών SSIM, PSNR και αριθμού βημάτων επανάληψης. Για τα πειράματά μας χρησιμοποιήσαμε ένα υποσύνολο περίπου 200 MRI εγκεφάλων.

Τα αποτελέσματα παρατίθενται στις γραφικές παραστάσεις 4.1.



(a) *Masked SSIM* (b) *Masked PSNR*

Figure 4.1: *Σύγκριση Αλγορίθμων - Κατακόρυφη Μάσκα*

Στις παραπάνω γραφικές, παρουσιάζουμε το μέσο SSIM και PSNR με κατακόρυφες μάσκες για διάφορες τιμές του παράγοντα επιτάχυνσης R. Οι σκιασμένες περιοχές αναπαριστούν διαστήματα 95% εμπιστοσύνης. Η μάσκα μηδενίζει όλες τις συντεταγμένες των οποίων η απόλυτη τιμή είναι μικρότερη από 0.05 φορές την μέγιστη απόλυτη τιμή που υπάρχει σε pixel της πλήρους δειγματοληφθείσας εικόνας, για να μειωθεί ο θόρυβος στο background. Επιπλέον, η μάσκα μηδενίζει όλα τα pixel εκτός κύκλου με κέντρο το κέντρο

της εικόνας και ακτίνα 165 pixel. Υψηλό R υποδεικνύει μικρότερο αριθμό μετρήσεων.

Ο SBIM αλγόριθμος με Langevin Dynamics corrector παράγει τα καλύτερα αποτελέσματα αναφορικά με την SSIM για όλα τα R, ενώ απαιτεί περίπου 15.5 φορές λιγότερες επαναλήψεις. Για R μικρότερο του 6, ο 32 βημάτων SBIM επίσης παράγει καλύτερα αποτελέσματα από τον Langevin Dynamics αναφορικά με την μετρική SSIM, ενώ είναι 125 φορές πιο γρήγορος.

Όσον αφορά την μετρική PSNR, ο καλύτερος SBIM με corrector παράγει τα πιο "καθαρά" δείγματα για R μικρότερο του 6. Για τις υπόλοιπες τιμές, ο Langevin Dynamics διατηρεί την πρωτιά στην μετρική PSNR, πιθανόν επειδή η διαδικασία αποθορυβοποίησής του έχει περισσότερο χρόνο για να ενσωματώσει τις (λιγοστές για μεγάλο R) μετρήσεις και να αφαιρέσει τον θόρυβο.

Τα σημαντικότερα αποτελέσματα παρατίθενται και στους πίνακες 4.1 και 4.2:

R/NFEs	32 (SBIM)	128 (SBIM)	128 (SBIM-3)	128 (LD)	3990 (LD)	256 (SBIM-3)
2	0.92	0.92	**0.93**	0.91	0.91	**0.93**
3	0.92	0.87	**0.93**	0.88	0.90	**0.93**
6	0.89	0.84	0.91	0.68	0.89	**0.92**
8	0.84	0.83	0.89	0.56	0.89	**0.90**
12	0.83	0.85	0.88	0.58	0.89	**0.90**

Table 4.1: *Σύγκριση Αλγορίθμων - SSIM - Κατακόρυφη Μάσκα*

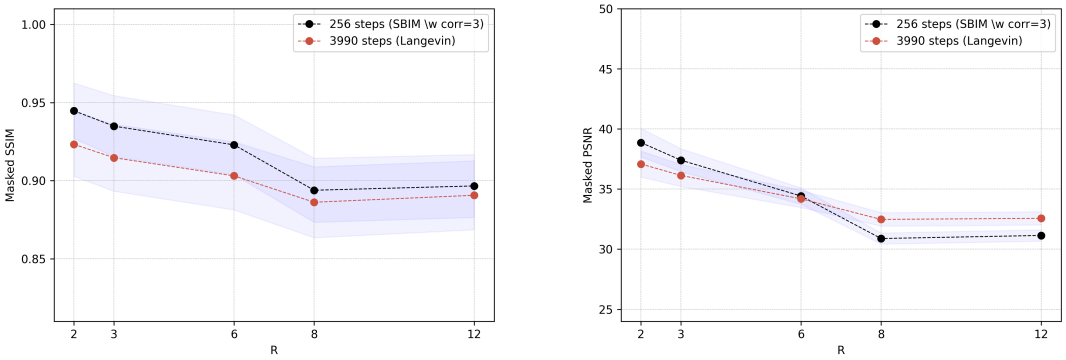
R/NFEs	32 (SBIM)	128 (SBIM)	128 (SBIM-3)	128 (LD)	3990 (LD)	256 (SBIM-3)
2	36.4	35.8	36.8	34.2	35.9	**37.1**
3	35.9	33.9	36.5	31.2	35.8	**36.8**
6	31.0	32.7	33.3	24.5	**34.1**	34.0
8	26.4	30.1	29.2	22.2	**32.9**	30.8
12	26.0	30.1	28.7	23.0	**32.3**	30.3

Table 4.2: *Σύγκριση Αλγορίθμων - PSNR - Κατακόρυφη Μάσκα*

Και στις δύο μετρικές ο SBIM (με διαφορετική για κάθε περίπτωση παραμετροποίηση βημάτων και corrector) παράγει πολύ καλά αποτελέσματα. Ο Langevin Dynamics είναι πολύ αργότερος του SBIM, ενώ ο δεύτερος υπολείπεται του πρώτου σε PSNR μόνο για μεγάλα R, κι αυτό όχι με μεγάλη απόκλιση. Ωστόσο, ο SBIM σε κάθε εκδοχή του είναι σημαντικά πιο γρήγορος από τον Langevin Dynamics.

Εντός κατανομής (εγκέφαλοι) - Οριζόντιες Μάσκες

Επαναλαμβάνοντας τα πειράματα για τους δύο καλύτερους δειγματολήπτες, δηλαδή τον 256 βημάτων SBIM με corrector και τον Langevin Dynamics, με οριζόντιο μασκάρισμα των MRI παίρνουμε παρόμοια αποτελέσματα. Οι διαρρυθμίσεις αναφορικά με τον υπολογισμό των SSIM και PSNR έχουν γίνει με τον ίδιο τρόπο, με την μόνη αλλαγή να είναι η οριζόντια μάσκα. Τα αποτελέσματα παρουσιάζονται στην Εικόνα 4.2 και στον Πίνακα 4.3.



(a) *Masked SSIM* (b) *Masked PSNR*

Figure 4.2: *Sampler Comparison - Horizontal Mask*

R/NFEs	256 (SBIM-3)	3990 (LD)
2	**0.94**	0.92
3	**0.93**	0.91
6	**0.92**	0.90
8	**0.89**	0.87
12	**0.90**	0.89

(a) *Masked SSIM*

R/NFEs	256 (SBIM-3)	3990 (LD)
2	**38.9**	37.1
3	**37.4**	36.1
6	**34.4**	34.2
8	30.9	**32.5**
12	31.14	**32.6**

(b) *Masked PSNR*

Table 4.3: *Sampler Comparison - Horizontal Mask*

Part II

Theoretical Background

Chapter 5

Introduction

A common, although very simplistic, division made in the Machine Learning field is between Discriminative and Generative modeling. In general, a discriminative model's job is to discriminate between different kinds of data instances based on learnt representations of the data distributions. On the other hand, a generative model plays the role of a "Con Artist" that creates brand new data aiming to synthesize them as indistinguishable from real ones as possible. A discriminative model could tell an apple from a banana, while a generative model can generate new images of realistic fruit. A more formal way to describe the aforementioned categories would be the following:

- Discriminative Models capture the probability distribution: $p(Y|X)$
- Generative Models capture the probability distribution: $p(X)$, or $p(X, Y)$ if dealing with labeled data.

Existing generative models can largely be grouped into two categories based on how they represent probability distributions [9].

- Likelihood-based models: trained to learn the distribution's probability density (or mass) function via (approximate) maximum likelihood. Autoregressive models [16, 17, 18], normalizing flow models [19, 20], energy-based models (EBMs) [21, 22], and variational autoencoders (VAEs) [23, 24] are typical examples of this category.
- Implicit generative models [25]: they represent the probability distribution via their sampling process. Generative adversarial networks (GANs) [26] are the most popular example; they generate new data by transforming a noise vector into them.

Both model categories suffer from significant limitations. Likelihood-based models either require strong restrictions on the model architecture to ensure a tractable normalizing constant for likelihood computation, or must rely on surrogate objectives to approximate maximum likelihood training [9]. Implicit generative models, on the other hand, often require adversarial training, which is notoriously unstable [27] and can lead to mode collapse [28, 9].

During the past three years a new family of generative models, namely Diffusion Models, has emerged and already become very popular in the field since they are able to

represent the distribution in a way that circumvents some of the aforementioned limitations (training stability, analytical tractability, architectural flexibility etc). The term "diffusion model" unifies two model families; the *denoising diffusion probabilistic models* [4] and the *score-based models* [29, 30, 3]. Both models gradually denoise a random image drawn from a known prior distribution into an image of a desired distribution, however their theoretical origins are generally different. The main drawback of these models is their sampling speed, which despite the many endeavors made in the field, still remains lower than GANs'.

Generative Models are a valuable tool for solving problems that require data distribution modeling, since they provide for black-boxes that contain implicit data representations. The prior distributions they capture come in handy when approaching signal processing *inverse problems*, such as inpainting, deblurring, denoising or compressed sensing, among others.

5.1 Thesis Objective

Compressed sensing [11] (also known as compressive sensing, compressive sampling, or sparse sampling) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. Recent progress in Deep Learning has shown that compressed sensing can be performed by using generative models as priors of the data distribution [12]. In this thesis, we are interested in magnetic resonance data images.

In the literature, two main generative approaches have been proposed:

1. The first approach uses generative models, namely GANs and VAEs, to capture the data distribution and gradient descent to optimize the latent representation $z \in R_k$ such that the corresponding image $G(z)$ has small measurement error $\|AG(z) - y\|^2$.

2. The second approach involves training a diffusion model (specifically, a score-based model) to capture the score function of the data distribution. Then, by employing iterative mathematical processes (posterior samplers), in which we incorporate the prior knowledge and measurements [12], we obtain excellent reconstructions of the MRI in terms of SSIM [31] and PSNR [32].

In this thesis, we investigate how we can reparameterize diffusion model samplers, such as DDIM, to employ score-based models instead and propose a derivative algorithm; Score-based implicit model (SBIM). SBIM is an algorithm that needs far less diffusion steps to produce samples, therefore it is of great interest to compare this method's performance against previous ones, such as Langevin Dynamics. During this endeavor, we propose SBIM as an alternative technique to solve linear inverse problems fast, as well a variation of SBIM, which also employs stochasticity. To evaluate our algorithms, we apply them to perform Compressed Sensing in MRI.

5.2 Thesis Organization

This thesis is organized in three parts; part I is an extensive summary of parts II and III in greek. Part II covers the theoretical background behind diffusion models, score-based models as well as the theoretical analysis that provides guarantees of model equivalence and posterior sampling convergence. In this part, the thesis refers to the connection between these models and corresponding stochastic differential equations. In the end of part II, compressed sensing in MRI is described and connected to diffusion-related techniques. Part III contains the theoretical reparameterization that we must apply upon a score-based model to "transform" it into a diffusion model during sampling time. This transformation leads to a new sampler, which we name Score-based Implicit Model (inspired by Denoising Diffusion Implicit Model), that allows faster sampling using score-based models. At last, we present how to utilize this derivation to accelerate a previous score-based framework to solve inverse problems and apply it to perform accelerated compressed sensing MRI. In the epilogue, the thesis contains references to related work on accelerated diffusion model sampling and a brief thesis summary.

Chapter 6

Score-based Generative Modeling

The family of diffusion models consists of two subgroups; denoising diffusion probabilistic models [4] and score-based models [29, 30, 3]. In this chapter, we focus on the latter subgroup. The key concept behind score-based models is to model a quantity that represents the data distribution, but is not the distribution itself. This quantity is the *gradient of the log probability density function*, known as the (Stein) *score function* [33].

Score-based models have achieved state-of-the-art performance on many downstream tasks, among which are image [29, 30, 4, 3, 34] and audio synthesis [35, 36, 37], shape [38] and music generation [39]. Furthermore, score-based models allow for exact likelihood computation and representation learning, and facilitate inverse problem solving.

The framework of score-based generative modeling consists of two key components; a score model and a sampler.

6.1 Networks that model gradients of the data distribution

Suppose our dataset consists of i.i.d. samples $\{x_i \in \mathbb{R}^D\}_{i=1}^N$ drawn from a data distribution $p_{data}(x)$. We define the score of a probability density $p(x)$ to be $\nabla_x log p(x)$. A score network $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$ is a neural network parameterized by θ, trained to approximate the score of the data distribution, i.e. $\nabla_x log p_{data}(x)$.

6.2 Score matching for score estimation

Despite its first-glance extravagance, $\nabla_x log p_{data}(x)$ can be learnt in simple ways under certain assumptions. The method used is known as *score matching* [6], and was originally designed for learning non-normalized statistical models based on i.i.d. samples from an unknown data distribution. Score matching for score estimation comes in two main variations; denoising score matching [7] and sliced score matching [8]. To learn the score function, we must minimize the following objective:

$$\frac{1}{2}\mathbb{E}_{p_{data}}\left[\|s_\theta(x) - \nabla_x log p_{data}(x)\|_2^2\right] \tag{6.1}$$

This is equivalent to the following up to a constant:

$$\mathbb{E}_{p_{data}}\left[\text{tr}(\nabla_x s_\partial(x)) + \frac{1}{2}\|s_\partial(x)\|_2^2\right] \tag{6.2}$$

where $\nabla_x s_\partial(x)$ denotes the Jacobian of $s_\partial(x)$. The expectation over $p_{data}(x)$ in Eq. (6.1) can be quickly estimated using data samples. However, the computation of $\text{tr}(\nabla_x s_\partial(x))$ does not allow score matching to be scalable to deep networks and high dimensional data [8]. To avoid this limitation we perform one of the following score matching techniques.

6.2.1 Denoising Score Matching

This variant of score matching [7] circumvents the computation of $\text{tr}(\nabla_x s_\partial(x))$. To perform this technique we first specify a noise distribution $q_\sigma(\tilde{x}|x)$ and perturb the data point x with it. The resulting (perturbed) distribution is $q_\sigma(\tilde{x}) \triangleq \int q_\sigma(\tilde{x}|x)p_{data}(x)\,dx$. Then, we estimate the score of the **perturbed** data distribution by minimizing the following:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{data}(x)}\left[\|s_\partial(\tilde{x}) - \nabla_{\tilde{x}} log\, q_\sigma(\tilde{x}|x)\|_2^2\right] \tag{6.3}$$

A neural net that minimizes Eq. (6.3) satisfies $s^*_\partial(x) = \nabla x log q_\sigma(x)$ almost surely [7]. However, $s^*_\partial(x) = \nabla_x log q_\sigma(x) \approx \nabla x log p_{data}(x)$ only when $\sigma \to 0$ (when $q_\sigma(x) \approx p_{data}(x)$).

6.2.2 Sliced Score Matching

Sliced score matching [8] estimates $\text{tr}(\nabla_x s_\partial(x))$ by using random projections. The objective is:

$$\mathbb{E}_{p_v}\mathbb{E}_{p_{data}}\left[v^T\nabla_x s_\partial(x)v + \frac{1}{2}\|s_\partial(x)\|_2^2\right] \tag{6.4}$$

where p_v is the multivariate standard normal. Forward mode auto-differentiation is used to efficiently compute $v^T\nabla_x s_\partial(x)v$ [8], but it requires around four times more computations than denoising score matching.

6.3 Langevin Dynamics

In score-based modeling, we want to use Langevin Dynamics as a sampler. This iterative procedure requires access to $\nabla_x log p(x)$ to produce samples, which explains our efforts to estimate the score function. Given a fixed step size and a starting point $\tilde{x}_0 \sim \pi(x)$ ($\pi(x)$ is a known prior), Langevin Dynamics can produce samples from $p_{\text{data}}(x)$ by making use of its score. It produces samples by iteratively computing the following:

$$\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x log p(\tilde{x}_{t-1}) + \sqrt{\epsilon}\, z_t \tag{6.5}$$

where $z_t \sim \mathcal{N}(0, I)$. When $\epsilon \to 0$ and $T \to \infty$, $\tilde{x}_T$ becomes an exact sample from $p(x)$ under some regularity conditions [40]. When $\epsilon > 0$ and $T < \infty$, the error of Eq. (6.5) is corrected by a Metropolis-Hastings update, but in practise, we set T to a large number and ϵ to a small number, and safely ignore this error.

6.4 Noise Conditional Score Networks (NCSNs)

Score-based generative modeling, faces two major challenges [29]; the manifold hypothesis and the low data density regions, both of which render score estimation not easily realizable.

In short, the manifold hypothesis states that most real-world datasets concentrate in low dimensional manifolds embedded in a high dimensional space (ambient space). In regions where no data live $\nabla_x logp(x)$ is not well defined. The low data density problem states that in regions where data is scarce $\nabla_x logp(x)$ cannot be accurately estimated.

To circumvent these challenges we perturb data points with noise and train score-based models on the noisy data points instead. When the noise used has large magnitude, it can populate the regions of data scarcity and mitigate the two problems.

When the noise is large it covers more low density regions and alleviates the challenges, but it severely corrupts the data distribution. Smaller noise, on the other hand, causes less corruption, but still suffers from the manifold hypothesis and the low data density.

To exploit the advantages of both cases, we perturb the data with multiple scales of noise perturbations at the same time [29, 30]. Let there be L fixed and increasing standard deviations $\sigma_1 < \sigma_2 < \cdots < \sigma_L$ and suppose that we always perturb the data with isotropic Gaussian noise $\mathcal{N}(0, \sigma_i I)$, using all $\sigma_i \in \{\sigma_i\}_{i=1}^L$. Then we obtain perturbed distributions: $q_{\sigma_i}(x) = \int p(y)\, \mathcal{N}(x; y, \sigma_i^2 I)\, dy$.

The resulting target for us is to estimate the scores of each perturbed distribution. But this is easily achievable, since we can apply score matching to estimate $\nabla_x logq_{\sigma_i}(x)$. Our way to capture this quantity is to employ a **Noise Conditional Score Network (NCSN)** $s_\partial(x, \sigma_i)$ parameterized by ∂, and train it to learn $s_\partial(x, \sigma_i) \approx \nabla_x logq_{\sigma_i}(x)$, $\forall i \in \{1, \ldots, L\}$.

6.5 Learning NCSNs via score matching

The most common method to estimate the score function of a distribution is via denoising score matching, since it naturally fits the task of estimating scores of noise-perturbed data distributions. For the perturbations we choose to utilize gaussian kernels $\mathcal{N}(0, \sigma^2 I)$. The noise distribution will be:

$$q_\sigma(\tilde{x}|x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left\{\frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right\} \tag{6.6}$$

and, consequently, the score will be equal to:

$$\nabla_x log\, q_\sigma(\tilde{x}|x) = \nabla_x\left\{log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right\} = -\frac{\tilde{x}-x}{\sigma^2} \tag{6.7}$$

Therefore, the denoising score matching objective (6.3) for a given σ becomes:

$$\ell(\partial, \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{data}}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2 I)}\left[\left\| s_\partial(\tilde{x}, \sigma) + \frac{\tilde{x}-x}{\sigma^2}\right\|^2\right] \tag{6.8}$$

Then, we combine Eq. (6.8) for all $\sigma \in \{\sigma_i\}_{i=1}^L$ to get one unified objective:

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^{L}) \triangleq \frac{1}{L} \sum_{i=1}^{L} \lambda(\sigma_i) \ell(\theta, \sigma_i) \tag{6.9}$$

There are many valid choices regarding the weighting factor λ, one of the most used ones being $\lambda(\sigma_i) = \sigma_i^a$, where a is called annealing power and is commonly set equal to 2.

6.6 NCSN inference via annealed Langevin Dynamics

To produce samples using our NCSN, the first method presented was annealed Langevin Dynamics, which is a modified version of classic Langevin Dynamics. This version is also iterative, but after each iteration the noise level σ_i is lowered and the ϵ factor is correspondingly decreased. The proposed sampling algorithm is (6.1).

ALGORITHM 6.1: *Annealed Langevin Dynamics*

Require: $\{\sigma\}_{i=1}^{L}, \epsilon, T$
1: Initialize $\tilde{x}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad a_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
4: $\quad$ **for** $i \leftarrow 1$ to T **do**
5: $\quad\quad z_t \sim \mathcal{N}(0, I)$
6: $\quad\quad \tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{a_i}{2} s_\theta(\tilde{x}_{t-1}, \sigma_i) + \sqrt{a_i}\, z_t$
7: $\quad \tilde{x}_0 \leftarrow \tilde{x}_T$
return $\tilde{x}_0$

In Figure 6.1, we present a visualization of three random sampling trajectories generated with Annealed Langevin dynamics [2], all starting from the same initialization point, for a Mixture of Gaussians. The left figure plots these sampling trajectories in 3D (hills represent distribution maxima) and the right in 2D (arrows represent the score function).

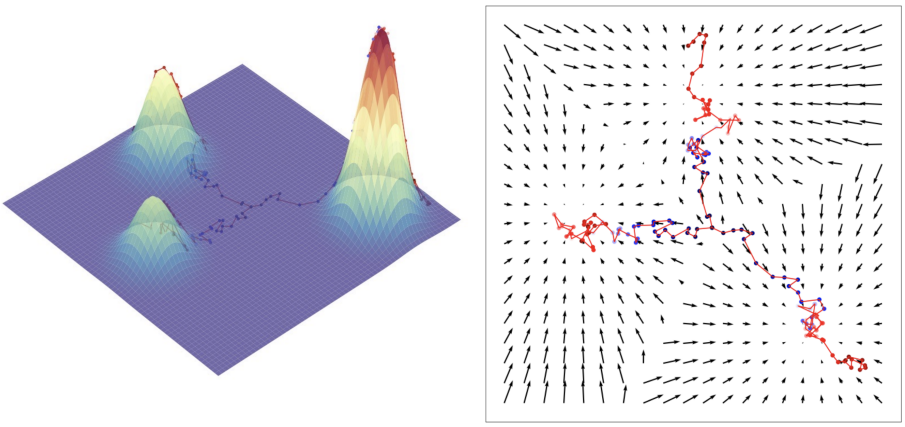


Figure 6.1: *Three random sampling trajectories generated with Langevin dynamics [2]*

Chapter 7

Diffusion Models

Diffusion models are sibling models of the score-based ones and were first proposed 4 years earlier (2015) [1]. Diffusion models are inspired by non-equilibrium thermodynamics. We define a Markov chain of diffusion steps and simulate them to gradually perturb the data with more noise. Then diffusion models are used to reverse the procedure of noise addition and, hence, produce realistic samples from noise. Diffusion models process the generated data while maintaining the dimensionality during the whole procedure. In general, diffusion models is an umbrella term and includes diffusion probabilistic models [1], noise conditional score models [29, 30], and denoising diffusion probabilistic models [4].

7.1 Why diffusion?

The idea behind diffusion models is the physics phenomenon of diffusion, i.e. the net movement of anything (for example, atoms, ions, molecules, energy) generally from a region of higher concentration to a region of lower concentration [41]. But how is that related to artificial generation of data?

Let's consider an infinite set of fragrances that exist in separate spaces, but they are released in ways that each fragrance has an initial spatial structure of a fruit. If we freeze the fragrances at this moment, we have infinite spaces each of whice contains a gas in the shape of a fruit. If we unfreeze this scene, then the fragrances will "naturally" start to diffuse and lose their shape. Each fragrance will be spatially distributed until it uniformly fills its space. The way (i.e. the number of shape transitions) that each gas loses its form is the phenomenon of diffusion, and it is characterized by randomness. The interesting part is not the way those gases lose their shape, but the idea of a stochastic reverse process. More specifically, what if we could model the reverse stochastic process of diffusion? We could have a randomly dispersed fragrance in a space that we could transform into the shape of a fruit!

The story behind diffusion models is how to model the reverse diffusion of a dataset that is "dispersed" into uniform shapes.

7.2 Forward diffusion process

Let $q(x)$ be a real distribution and $x_0 \sim q(x)$. Also, let us define a schedule of noise variances $\{\beta_t \in (0,1)\}_{t=0}^{T}$. We define the forward diffusion process to be the gradual addition of noise to a clean datum x_0, such that at each discrete timestep t the noise perturbing the datum is an isotropic Gaussian with variance β_t. When $T \to \infty$, as t increases, the datum loses its distinguishable features and becomes pure isotropic Gaussian sample. The Markov chain is defined by the following diffusion kernel:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{7.1}$$

And the joint distribution, which is called the *forward diffusion process* is:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{7.2}$$

The forward process is easily implemented, since we have knowledge of every distribution related to it. The reverse procedure is not. The purpose of diffusion models is to capture the distribution of the reverse process $q(x_{t-1}|x_t)$, which is unknown.

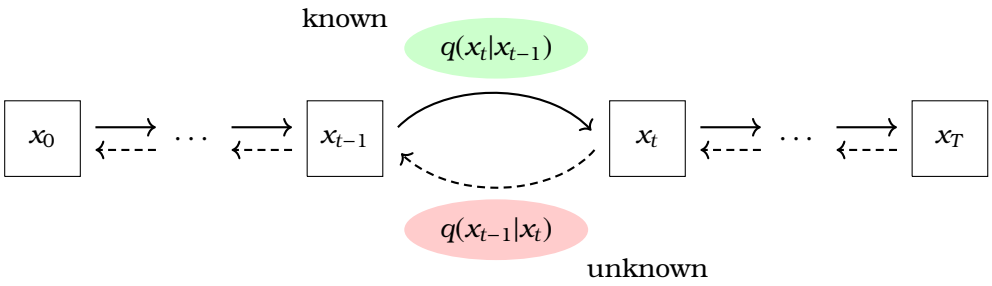


Figure 7.1: *Forward and Reverse Diffusion Processes*

Lemma 7.1. *Let z be a random variable drawn from a multivariate Gaussian with a diagonal covariance structure, i.e. $z \sim q(z|x) = \mathcal{N}(z; x, \sigma^2 I)$. Then, z can be equivalently expressed in the form:*

$$z = x + \sigma \odot \epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\odot$ denotes element-wise multiplication.

Proposition 7.2. *Let x_t be drawn from $q(x_t|x_{t-1})$ at timestep t. Then x_t can be directly expressed as:*

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$

Proof. Let $t \in Cat[0, 1, 2, \ldots, T]$. Then, by Eq. (7.1): $x_t \sim q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t\, I)$. Now, let $a_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} a_i$.

According to Lemma 7.1 and our new notation, we can express $x_t = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \sqrt{\beta_t}\, I)$

as follows:

$$\begin{aligned} x_t &= \sqrt{a_t}\, x_{t-1} + \sqrt{1-a_t}\, \epsilon_{t-1} && \text{where } \epsilon_{t-1} \sim \mathcal{N}(0, I) \\ &= \sqrt{a_t}\left[\sqrt{a_{t-1}}\, x_{t-2} + \sqrt{1-a_{t-1}}\, \epsilon_{t-2}\right] + \sqrt{1-a_t}\, \epsilon_{t-1} && \text{where } \epsilon_{t-2} \sim \mathcal{N}(0, I) \\ &= \sqrt{a_t a_{t-1}}\, x_{t-2} + \underbrace{\sqrt{a_t - a_t a_{t-1}}\, \epsilon_{t-2} + \sqrt{1-a_t}\, \epsilon_{t-1}} \end{aligned}$$

The underbraced quantity is the sum of two gaussian distributions with different variances, $\mathcal{N}(0, \sigma_1^2 I)$ and $\mathcal{N}(0, \sigma_2^2 I)$. This sum can be expressed as: $\mathcal{N}(0, (\sigma_1^2 + \sigma_2^2) I)$. Therefore, our last equation can be expressed as follows:

$$x_t = \sqrt{a_t a_{t-1}}\, x_{t-2} + \sqrt{1 - a_t a_{t-1}}\, \bar{\epsilon}_{t-2} \qquad \text{where } \bar{\epsilon}_{t-2} \sim \mathcal{N}(0, I)$$

By repeating the above thinking, one sees that x_t can take this final form:

$$x_t = \sqrt{\bar{a}_t}\, x_0 + \sqrt{1 - \bar{a}_t}\, \epsilon \qquad \text{where } \epsilon \sim \mathcal{N}(0, I)$$

□

At this point, it has become obvious that **sampling from any intermediate distribution of the forward diffusion process will require a single step**, since (according to proposition 7.2) the distributions $\mathcal{N}(x_t; \sqrt{a_t}\, x_{t-1}, (1 - a_t) I)$ and $\mathcal{N}(x_t; \sqrt{\bar{a}_t}\, x_0, (1 - \bar{a}_t) I)$ are equivalent.

7.3 Reverse diffusion process

The interesting part of diffusion is the reverse process. If one could model the reverse diffusion, then it could be possible to produce a sample from the data distribution by applying reverse diffusion on an initial Gaussian sample $x_T \sim \mathcal{N}(0, I)$. It is also known that if β_i is small enough, then $q(x_{t-1|x_t})$ will also be Gaussian [42]. To perform reverse diffusion, we need to use this observation and model the kernels of the reverse timesteps. To estimate $q(x_{t-1|x_t})$ it would need to use the entire dataset, which is impractical, and therefore we need to learn a model p_∂ to approximate these conditional probabilities. The joint reverse distribution and revere diffusion kernels are:

$$p_\partial(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\partial(x_{t-1}|x_t) \quad p_\partial(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\partial(x_t, t), \Sigma_\partial(x_t, t)) \tag{7.3}$$

This far, $q(x_{t-1}|q_t)$ is not tractable. However, if we condition it on x_0, then it becomes tractable:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I})$$

We apply Bayes' rule:

$$
\begin{aligned}
q(x_{t-1}|x_t, x_0) &= q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&\propto \exp\Big[-\frac{1}{2}\Big(\frac{(x_t - \sqrt{a_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{a}_{t-1}}x_0)^2}{1-\bar{a}_{t-1}} - \underbrace{\frac{(x_t - \sqrt{\bar{a}_t}x_0)^2}{1-\bar{a}_t}}_{C'(x_t, x_0)}\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\frac{x_t^2 - 2\sqrt{a_t}x_t x_{t-1} + a_t x_{t-1}^2}{\beta_t} + \frac{x_{t-1}^2 - 2\sqrt{\bar{a}_{t-1}}x_0 x_{t-1} + \bar{a}_{t-1}x_0^2}{1-\bar{a}_{t-1}} - C'\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\underbrace{\Big(\frac{a_t}{\beta_t} + \frac{1}{1-\bar{a}_{t-1}}\Big)}_{A}x_{t-1}^2 - \underbrace{\Big(\frac{2\sqrt{a_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{a}_{t-1}}}{1-\bar{a}_{t-1}}x_0\Big)}_{B}x_{t-1} + \underbrace{C(x_t, x_0)}_{\Gamma}\Big)\Big]
\end{aligned}
$$

Where $C(x_t, x_0)$ is a function not involving x_{t-1}.

$$
\begin{aligned}
&= \exp\Big[-\frac{1}{2}(A\,x_{t-1}^2 - B\,x_{t-1} + \Gamma)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\sqrt{A}\,x_{t-1} - \frac{B}{2\sqrt{A}}\Big)^2 - \Big(\frac{B}{2\sqrt{A}}\Big)^2 + \Gamma\Big] \\
&\propto \exp\Big[-\frac{1}{2}\Big(\sqrt{A}\,x_{t-1} - \frac{B}{2\sqrt{A}}\Big)^2\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\frac{x_{t-1} - \frac{B}{2A}}{\frac{1}{\sqrt{A}}}\Big)^2\Big]
\end{aligned}
$$

Since $q(x_{t-1}|x_t, x_0)$ is Gaussian, it is now clear that:

$$
\tilde{\mu}(x_t, x_0) = \frac{\sqrt{a_t}(1-\bar{a}_{t-1})}{1-\bar{a}_t}x_t + \frac{\sqrt{\bar{a}_{t-1}}\beta_t}{1-\bar{a}_t}x_0
$$

$$
\tilde{\beta}_t = \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\beta_t
$$

By making use of Lemma 7.1, we can express x_0 in the form:

$$
x_0 = \frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1-\bar{a}_t}z_t)
$$

Plugging this quantity back into the previous equation, we get:

$$
\begin{aligned}
\tilde{\mu}_t &= \frac{\sqrt{a_t}(1-\bar{a}_{t-1})}{1-\bar{a}_t}x_t + \frac{\sqrt{\bar{a}_{t-1}}\beta_t}{1-\bar{a}_t}\frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1-\bar{a}_t}z_t) \\
&= \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}z_t\Big)
\end{aligned}
$$

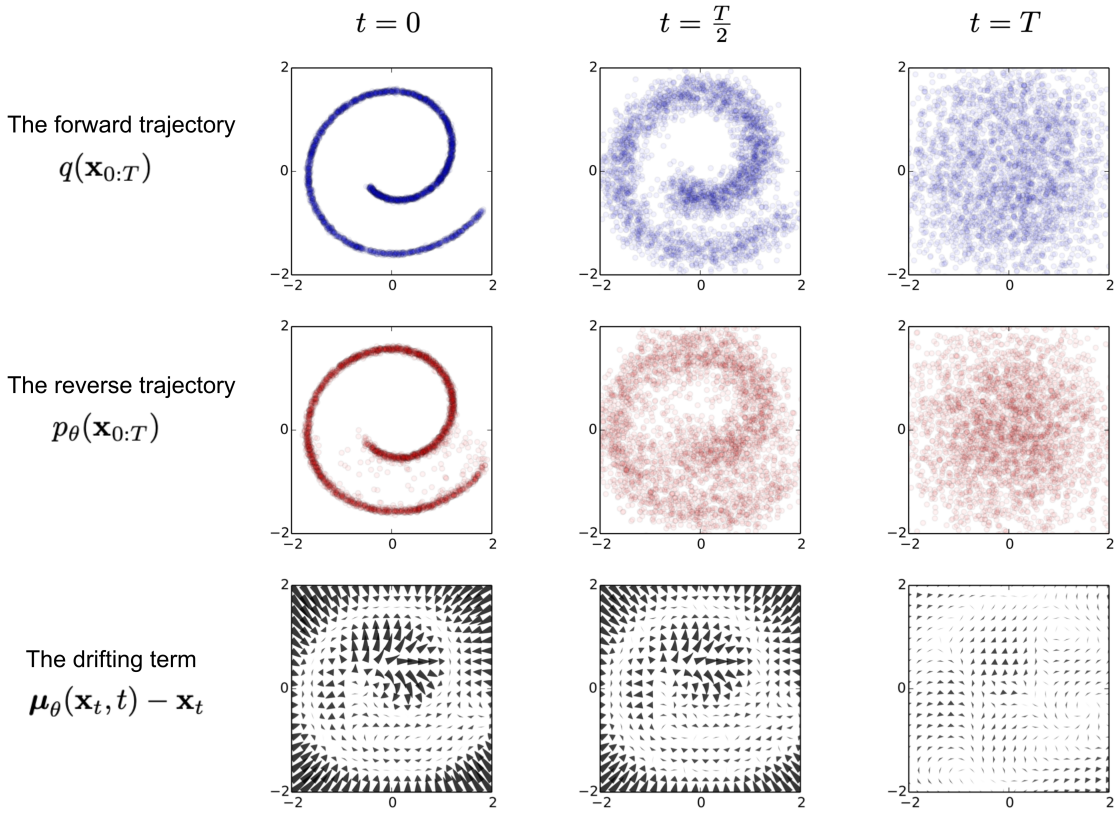


Figure 7.2: *Training a diffusion model for modeling a 2D swiss roll data. [1]*

7.4 Parameterization of the training objective

Our target is to employ a diffusion model to simulate the reverse diffusion by approximating the reverse diffusion kernels $p_\partial(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\partial(x_t, t), \Sigma_\partial(x_t, t))$. The model is going to help us to approximate $\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_t\Big)$. We have access to x_t at training time, and we train the model to predict the Gaussian noise term from the input x_t at timestep t.

$$\mu_\partial(x_t, t) = \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_\partial(x_t, t)\Big)$$
$$\text{Thus } x_{t-1} = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_\partial(x_t, t)\Big), \Sigma_\partial(x_t, t))$$

The loss term L_t is parameterized to minimize the difference from $\tilde{\mu}$:

$$\begin{aligned}
L_t &= \mathbb{E}_{x_0,z}\Big[\frac{1}{2\|\Sigma_\partial(x_t,t)\|_2^2}\|\tilde{\mu}_t(x_t, x_0) - \mu_\partial(x_t, t)\|^2\Big] \\
&= \mathbb{E}_{x_0,z}\Big[\frac{1}{2\|\Sigma_\partial\|_2^2}\|\frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_t\Big) - \frac{1}{\sqrt{a_t}}\Big(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} z_\partial(x_t, t)\Big)\|^2\Big] \\
&= \mathbb{E}_{x_0,z}\Big[\frac{(1-a_t)^2}{2a_t(1-\bar{a}_t)\|\Sigma_\partial\|_2^2}\|z_t - z_\partial(x_t, t)\|^2\Big] \\
&= \mathbb{E}_{x_0,z}\Big[\frac{(1-a_t)^2}{2a_t(1-\bar{a}_t)\|\Sigma_\partial\|_2^2}\|z_t - z_\partial(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}z_t, t)\|^2\Big]
\end{aligned}$$

7.4.1 Training loss simplification

Ho et al. [4] empirically found that a simplified version of the training objective that ignores the weighting term works better at training a diffusion model:

$$\begin{aligned} L_t^{\text{simple}} &= \mathbb{E}_{t\sim[1,T],x_0,z_t}\Big[\|z_t - z_\partial(x_t, t)\|^2\Big] \\ &= \mathbb{E}_{t\sim[1,T],x_0,z_t}\Big[\|z_t - z_\partial(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}z_t, t)\|^2\Big] \end{aligned}$$

The final simple objective is:

$$L_{\text{simple}} = L_t^{\text{simple}} + C$$

where C is a is a constant not depending on ∂.

7.5 Denoising Diffusion Implicit Models

So far, all candidate samplers are stochastic, i.e. there is always noise injected throughout the reverse diffusion process. Since the reverse diffusion is a Markovian chain, each step relies on its previous one and in order to obtain good results it is most probable that one is going to have to perform all intermediate steps. However, it is very slow to generate a sample from DDPM by following the Markov chain of the reverse diffusion process, as T can be up to a few thousand steps.

A simple suggestion would be to run a strided sampling schedule [43] by taking the sampling update every $[T/S]$ steps to reduce the process from T to S steps. The new sampling schedule for generation is $\{\tau_1, \dots, \tau_S\}$ where $\tau_1 < \tau_2 < \cdots < \tau_S \in [1, T]$ and $S < T$. This scheme, however, is most definitely going to damage the sampling quality.

The first approach to significantly accelerate sampling is called Denoising Diffusion Implicit Model [5] (DDIM) and is the first sampler that can generate samples in a **deterministic** fashion, while maintaining the sampling quality.

7.5.1 Sampler derivation

For this approach, let us reparameterize the reverse diffusion distributions q_σ, according to Lemma 7.1:

$$\begin{aligned} x_{t-1} &= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}}z_{t-1} \\ &= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}z_t + \sigma_t z \\ &= \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}\frac{x_t - \sqrt{\bar{a}_t}x_0}{\sqrt{1-\bar{a}_t}} + \sigma_t z \\ q_\sigma(x_{t-1}|x_t, x_0) &= \mathcal{N}(x_{t-1};\ \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1-\bar{a}_{t-1}-\sigma_t^2}\frac{x_t - \sqrt{\bar{a}_t}x_0}{\sqrt{1-\bar{a}_t}},\ \sigma_t^2\mathbf{I}) \end{aligned}$$

Recall that in $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I})$, therefore we have:

$$\tilde{\beta}_t = \sigma_t^2 = \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\cdot\beta_t$$

We now set $\sigma_t^2 = \eta \cdot \tilde{\beta}_t, \eta \in \mathbb{R}^+$. We observe that this setting allows us to adjust the stochasticity of the sampling procedure via controlling the η hyperparameter. If we set $\eta = 0$, then the sampling becomes deterministic, and this sampler's name is Denoising Diffusion Implicit Model (DDIM) [5], since it simulates what a denoising diffusion probabilistic model does. The word *implicit* refers to the fact that the forward procedure that corresponds to DDIM's reverse diffusion is no longer a diffusion.

During generation, we only sample a subset of S diffusion steps $\{\tau_1, \ldots, \tau_S\}$ and the inference process becomes:

$$q_{\sigma,\tau}(x_{\tau_{i-1}}|x_{\tau_t}, x_0) = \mathcal{N}(x_{\tau_{i-1}};\ \sqrt{\bar{a}_{t-1}}x_0 + \sqrt{1 - \bar{a}_{t-1} - \sigma_t^2}\frac{x_{\tau_i} - \sqrt{\bar{a}_t}x_0}{\sqrt{1 - \bar{a}_t}}, \sigma_t^2\mathbf{I})$$

In practise, it has been observed that DDIM performs better when executed on a subset of the initial T diffusion steps that the model was trained on, while DDPM (ancestral sampling) produces the best results when applied on all T initial steps. With DDIM, it is possible to train the diffusion model up to any arbitrary number of forward steps but only sample from a subset of steps in the generative process.

7.6 Diffusion Models Summary

Given the mathematical proofs provided in the previous sections, the algorithms one needs to train a diffusion model and utilize it to produce samples from a data distribution are the following three:

ALGORITHM 7.1: *Training diffusion models [4]*

repeat
  $x_0 \sim q(x_0)$
  $t \sim \text{Uniform}(\{1, \ldots, T\})$
  $\epsilon \sim \mathcal{N}(0, I)$
  Take gradient descent step on: $\nabla_\theta \|\epsilon - z_\theta(\sqrt{\bar{a}_t}x_0 + \sqrt{1 - \bar{a}_t}\epsilon, t)\|^2$
until converged

ALGORITHM 7.2: *Ancestral sampling [4]*

$x_0 \sim \mathcal{N}(0, I)$
for $t \leftarrow T$ to 1 **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$ else $z = 0$
  $x_{t-1} \leftarrow \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}z_\theta(x_t, t)\right) + \sigma_t z$
return x_0

ALGORITHM 7.3: *Denoising Diffusion Implicit Model [5]*

$x_T \sim \mathcal{N}(0, I)$
for $t \leftarrow T$ to 1 **do**
  $x_{t-1} \leftarrow \sqrt{\bar{a}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{a}_t}\epsilon_\theta(x_t)}{\sqrt{\bar{a}_t}}\right) + \sqrt{1 - \bar{a}_{t-1}}\epsilon_\theta(x_t)$
return x_0

Chapter 8

Score-based Generative Modeling with Stochastic Differential Equations

Score-based generative models and diffusion models gradually perturb data with more noise and then learn how to reverse this process, in order to generate new data from random noise. More specifically, score-based models learn the score functions (i.e., the gradient of the log probability density with respect to data) of distributions that result from perturbing clean data with noise of pre-specified standard deviations, and then perform Annealed Langevin Dynamics to sample from the reverse sequence of decreasing noise scales. On the other hand, denoising diffusion probabilistic models (DDPM) [1, 4] are trained to reverse each step of the noise corruption. Song et al. [3] unified previous approaches by showing that both types of generation are discretizations of stochastic differential equations (SDEs).

In this framework, instead of perturbing the data with a fixed number of noise distributions, an infinite set of distributions is considered. The distributions of this set are defined by continuous noise scales. The continuous nature of the noise perturbation generalizes the data corruption into a continuous diffusion process given by a prescribed SDE (that is known and fixed). Each SDE (called the forward SDE) has a closed-form reverse-time SDE [44] that depends on the score of the marginal probability densities as a function of time. By solving the reverse-time SDE, we can generate data from noise that evolves into them following a reverse-SDE trajectory. To approximate this reverse evolution we train a time-dependent neural network to estimate the scores, and then generate new samples by utilizing SDE solvers.

8.1 Perturbing data with an SDE

Previously, the key idea of score-based generative modeling was to perturb the data with a finite set of noise distributions. This idea is generalized through the lens of SDEs, where we perturb the data with infinite noise distributions, such that the perturbed data distributions evolve according to a predefined SDE. A high-level overview of this idea is visualized in Fig. 8.1.

Diffusion processes [41] (among other stochastic processes) are, in fact, solutions to stochastic differential equations (SDEs). One general form describing an SDE is the

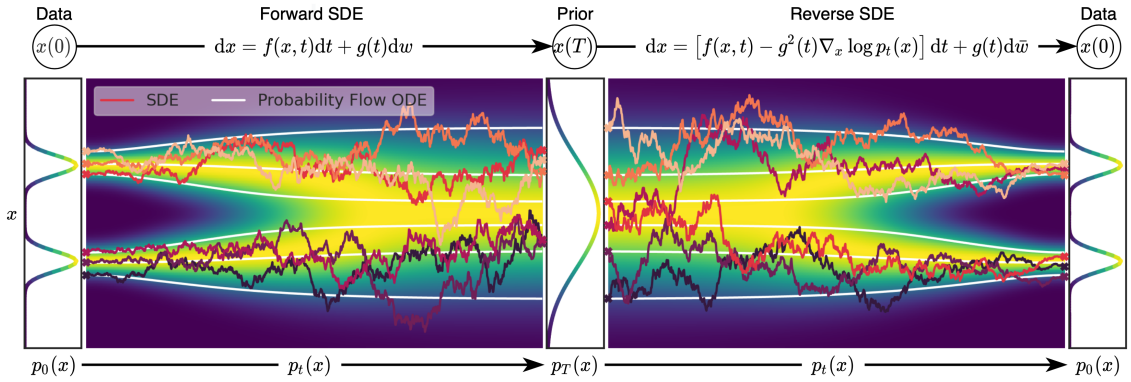


Figure 8.1: *Overview of score-based generative modeling through SDEs [3]*

following:

$$dx = f(x,t)dt + g(t)dw \tag{8.1}$$

where $f(\cdot,t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function called the drift coefficient, $g(t) \in \mathbb{R}$ is a real-valued function called the diffusion coefficient, w denotes a standard Brownian motion [10] and dw can be viewed as infinitesimal white noise. The solution of a stochastic differential equation is a continuous collection of random variables $\{x(t)\}_{t\in[0,T]}$. These random variables trace stochastic trajectories as the time index t grows from the start time 0 to the end time T.

Now we describe the analogy between the continuous (generalization) and the discrete (previous method of finite noise scales) cases of diffusion. The discrete steps $i = 1, 2, \ldots, L$ are now generalized to continuous timesteps $t \in [0, T]$. Let $p(x,t)$ be the marginal probability density function of $x(t)$ at the timesteps t. Then $p(x,0)$ is the same as $p_0(x)$ from the discrete case and is the *clean* data distribution. Also, $p(x,T)$ is the same as $p_T(x)$ (both are close to a tractable noise distribution) and corresponds to applying the largest noise perturbation σ_L to the data.

The noise scales $\sigma_1, \sigma_2, \ldots, \sigma_L$ in the discrete case are hand designed, and, in general, there are numerous was to choose them. Similarly, the SDE in 8.1 is also hand-designed and varies in accordance to the change we make to the noise scales of the discrete case. In other words, as we change discrete and finite set of noise scales, so does the corresponding SDE that generalizes it. For example, the case of choosing the noise schedule $\{sigma_i\}_{i=1}^L$ to be a geometric progression can be described by the following forward SDE:

$$dx = e^t dw \tag{8.2}$$

This SDE pertubs the data with Gaussian noise of mean zero and exponentially growing variance, which is analogous to perturbing the data with $\mathcal{N}(0, \sigma_1^2 I), \ldots, \mathcal{N}(0, \sigma_L^2 I)$ when $\{\sigma_i\}_{i=1}^L$ is a geometric progression. This means that this SDE, in fact, generalizes the case of discrete noise scales that increase in a geometric progression. Much like the noise scales that a score-based model uses are a part of the model, in the continuous case **the SDE is part of the model**. Two very important SDEs that work well with images and

naturally fit the generating methods of score-based models and diffusion models are the **Variance Exploding (VE)** and the **Variance Preserving (VP)** SDEs, correspondingly.

8.2 Variance Exploding and Variance Preseving SDEs

The noise perturbations of score-based models and diffusion models are discretizations of two different SDEs ([3], Appendix B).

Let N be the noise scales we use to perturb the data. For the score-based models' case, each perturbation kernel $p_{\sigma_i}(x|x_0)$ is derived from the following Markov chain:

$$x_t = x_{t-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, \dots, N \tag{8.3}$$

where z_i is random Gaussian noise of zero mean and unit variance (σ_0 is used for simplification). When the noise scales are infinite ($N \to \infty$), the set of discrete $\{\sigma_i\}_{i=1}^{\infty}$ becomes a continuous function $\sigma(t)$ and the noise z_i becomes $z(t)$. Also, the Markov chain $\{x_i\}_{i=1}^{T}$ becomes a continuous-time stochastic process $\{x(t)\}_{t=0}^{1}$. The corresponding SDE whose solution is the process $\{x(t)\}_{t=0}^{1}$ is:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw \tag{8.4}$$

Likewise for the perturbation kernels $\{p_{\alpha_i}(x|x_0)\}_{i=1}^{N}$ of diffusion models, the discrete Markov chain is:

$$x_i = \sqrt{1-\beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}, \quad i = 1, \dots, N \tag{8.5}$$

As $N \to \infty$, Eq. 8.5 converges to the following SDE:

$$dx = -\frac{1}{2}\beta(t)x\, dt + \sqrt{\beta(t)}\, dw \tag{8.6}$$

Based on the above, it is clear that the discrete noise perturbations of score-based models are a discretization of Eq. 8.4, while those of the diffusion models are a discretization of Eq. 8.6. The solution to the SDE of Eq. 8.4 is a process with exploding variance when $t \to \infty$. On the other hand, the solution to the SDE of Eq. 8.6 yields a process with a fixed variance of one when the initial distribution has unit variance (proof in [3], Appendix B). Inspired from these characteristics, Eq. 8.4 is known as the Variance Exploding (VE) SDE, and Eq. 8.6 as the Variance Preserving (VP) SDE.

8.3 Reversing the SDE for sample generation

Recall that when the number of noise scales is finite, then we can generate samples via (for example) Annealed Langevin Dynamics, which reverses the noise perturbation process. When we deal with the continuous case, we also generate samples by reversing the perturbation process, by making use of the reverse SDE.

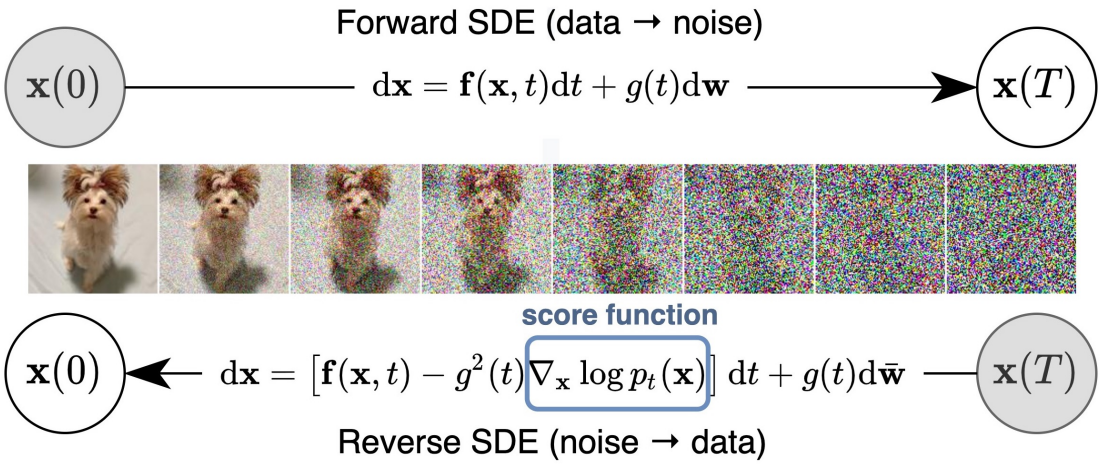


Figure 8.2: *Solving a reverse SDE yields a score-based generative model [3]*

Given the notation used for the forward SDE, the corresponding reverse SDE (which always exists [44]), has the following closed form:

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dw \tag{8.7}$$

This SDE is to be solved backwards in time, hence dt represents a negative infinitesimal time step. In order to compute the reverse SDE, we need to estimate $\nabla_x \log p_t(x)$, which is exactly the score function that score-based models are trained to learn.

We sample $x(T)$ from a known prior distribution (most commonly being Gaussian), and solve the above reverse SDE (backwards in time) to obtain $x(0)$, whose distribution is approximately equal to the distribution of clean data (when accessing the score-function via an optimal model).

8.3.1 General purpose numerical SDE solvers

To solve the reverse SDE, we can employ numerical SDE solvers, which provide approximate SDE trajectories. One of the most popular SDE solvers is the Euler-Maruyama method [45], which corresponds to different discretizations of the stochastic dynamics. In general, we can apply any SDE solver to reverse the SDE and produce samples.

8.3.2 Predictor-Corrector samplers

Predictor-corrector (PC) samplers are 2-component iterative algorithms that make a "prediction" and a "correction" at each step. The first component is an estimate of the sample at the next time step, made by the **predictor**, and the second component is a correction of the marginal distribution of the estimated sample, made by the **corrector**. The predictor is a numerical SDE solver, while the corrector is a score-based MCMC approach, such as Langevin Dynamics.

The proposed predictor-corrector algorithms as proposed in [3] are given below:

ALGORITHM 8.1: *PC Sampling (VE SDE)*

$x_N \sim \mathcal{N}(0, \sigma_{max}^2 I)$
for $i \leftarrow N-1$ to 0 **do**
  $x_i' \leftarrow x_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) s_\partial * (x_{i+1}, \sigma_{i+1})$
  $z \sim \mathcal{N}(0, I)$
  $x_i \leftarrow x_i' + \sqrt{s_{i+1}^2 - \sigma_i^2}\, z$
  for $j \leftarrow 1$ to M **do**
    $z \sim \mathcal{N}(0, I)$
    $x_i \leftarrow x_i + \epsilon_i s_\partial^*(x_i, \sigma_i) + \sqrt{2\epsilon_i}\, z$
return x_0

ALGORITHM 8.2: *PC Sampling (VP SDE)*

$x_N \sim \mathcal{N}(0, I)$
for $i \leftarrow N-1$ to 0 **do**
  $x_i' \leftarrow (2 - \sqrt{1-\beta_{i+1}})x_{i+1} + \beta_{i+1} s_\partial^*(x_{i+1}, i+1)$
  $z \sim \mathcal{N}(0, I)$
  $x_i \leftarrow x_i' + \sqrt{\beta_{i+1}}\, z$
  for $j \leftarrow 1$ to M **do**
    $z \sim \mathcal{N}(0, I)$
    $x_i \leftarrow x_i + \epsilon_i s_\partial^*(x_i, \sigma_i) + \sqrt{2\epsilon_i}\, z$
return x_0

ALGORITHM 8.3: *Corrector Algorithm (VE)*

Require: $\{\sigma_i\}_{i=1}^N, r, N, M$
$x_N^0 \sim \mathcal{N}(0, \sigma_{max}^2 I)$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $z \sim \mathcal{N}(0, I)$
    $g \leftarrow s_\partial^*(x_i^{j-1}, \sigma_i)$
    $\epsilon \leftarrow 2\left(\frac{r\|z\|_2}{\|g\|_2}\right)^2$
    $x_i^j \leftarrow x_i^{j-1} + \epsilon\, g + \sqrt{2\epsilon}\, z$
  $x_{i-1}^0 \leftarrow x_i^M$
return x_0^0

ALGORITHM 8.4: *Corrector Algorithm (VP)*

Require: $\{\beta_i\}_{i=1}^N, r, N, M$
$x_N^0 \sim \mathcal{N}(0, I)$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $z \sim \mathcal{N}(0, I)$
    $g \leftarrow s_\partial^*(x_i^{j-1}, i)$
    $\epsilon \leftarrow 2a_i\left(\frac{r\|z\|_2}{\|g\|_2}\right)^2$
    $x_i^j \leftarrow x_i^{j-1} + \epsilon\, g + \sqrt{2\epsilon}\, z$
  $x_{i-1}^0 \leftarrow x_i^M$
return x_0^0

Chapter 9

Compressed Sensing

Compressed sensing [11], which may also be referred to as Compressive Sensing, Compressive Sampling or Sparse Sampling, is a method used to efficiently acquire and reconstruct signals, which have been sampled at a lower rate than the Nyquist frequency threshold. In a more mathematical sense, compressed sensing is a signal processing technique for finding solutions to underdetermined linear systems.

Its main idea is based on the fact that many real world signals tend to possess useful properties which can be exploited to recover them from far fewer samples than required by the Nyquist–Shannon sampling theorem. There are two conditions under which recovery is possible; the first one is sparsity, which requires the signal to be sparse in some domain (the DCT coefficients domain, for instance). The second one is incoherence, which is applied through the isometric property, which is sufficient for sparse signals. To develop intuition of the usefulness and mechanics of compressed sensing, Fig. 9.2 provides a detailed example of its application to a toy signal.

Compressed sensing is related to several topics in signal processing and computational mathematics and is applied in various domains such as photography, holography, facial recognition, magnetic resonance imaging (MRI) and network tomography, among others.

9.1 Methodology

9.1.1 Underdetermined Linear Systems

By definition, an underdetermined linear system is a system of equations that has fewer equations than unknown variables. Those systems have an infinite number of possible solutions.

To find a solution to such a system that is appropriate for one's uses, it is necessary to impose extra constraints or conditions. Compressed sensing is a technique that allows the recovery of a *sparse* solution, i.e. has a very low number of non-zero elements. Not all underdetermined systems have a sparse solution, however if a unique sparse solution exists, then the compressed sensing framework will allow one to find it.

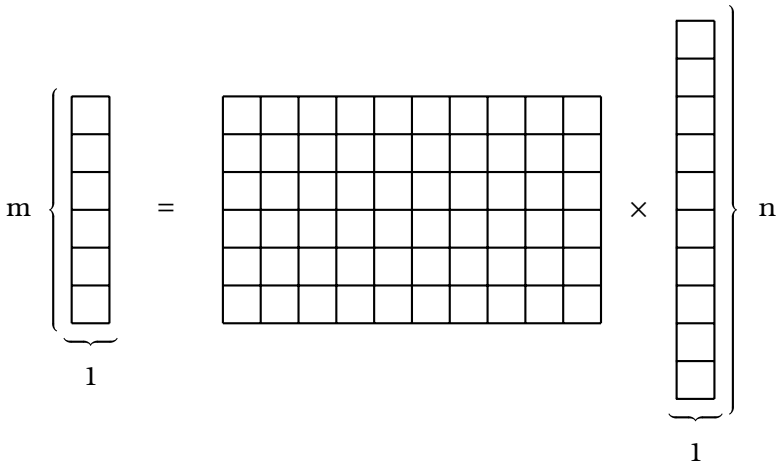


Figure 9.1: *Underdetermined Linear System:* $x = A \cdot y$

9.1.2 General Solution Method

Many interesting signals are characterized by an information redundancy, i.e. they carry more information than needed so as to be represented concisely. Signals like these are not pure noise and, when represented in a specific domain, they are sparse (they contain many coefficients close to or equal to zero).

The first step to perform Compressed Sensing is to take the **compressive measurements**. These measurements are a weighted linear combination of samples in a basis different from the one in which the signal of interest is known to be sparse [11]. Related research has shown that the compressive measurements can be very few, but all useful information can still be contained within them. The measurements have much lower dimensionality than the original signal, hence to convert the image back into the intended domain, one must solve an underdetermined matrix equation. By adding the constraint of sparsity on the original signal, however enables system solution.

One idea would be to apply the least-squares approach, which minimizes the L_2 norm (equivalent to minimizing the system's energy). This approach is usually simple to perform mathematically, since it involves only a matrix multiplication by the pseudo-inverse of the basis in which sampling was performed. Despite the appeal of its simplicity, this technique is not preferred in practise, in as much as it produces poor results when the unknown coefficients have nonzero energy.

The sparsity property can be enforced by minimizing the L_0 norm of the solution, which is a function counting the number of non-zero components of a vector. It is proven that for many problems, the L_1 and L_0 norms are equivalent with high probability, allowing for the latter's minimization via solving a problem that minimizes the former. This mitigates the problem of L_0 minimization, which is generally more difficult. Finding the candidate with the smallest L_1 norm can be expressed relatively easily as a linear program, for which efficient solution methods already exist [46].

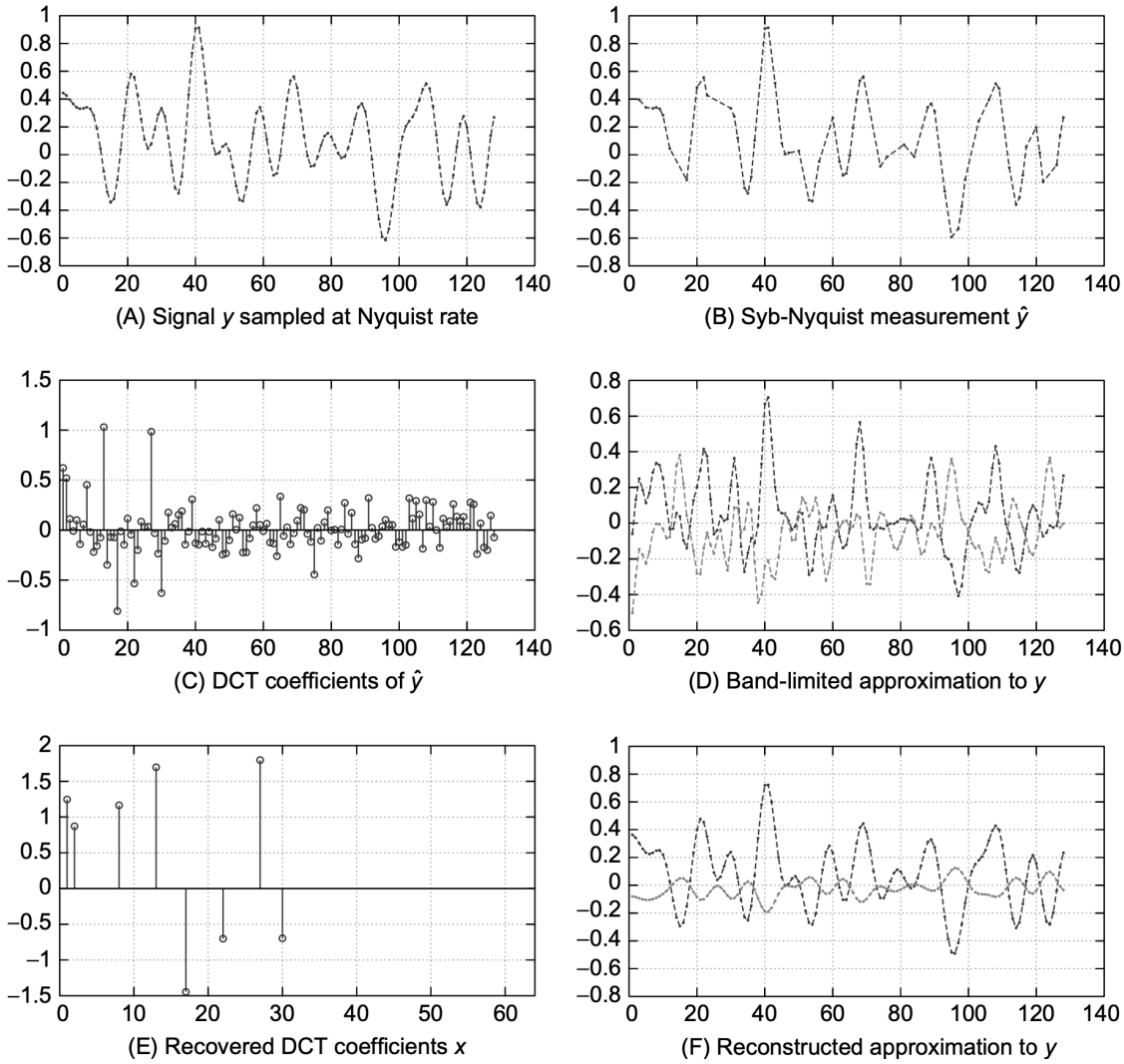


Figure 9.2: *Toy example of Compressed Sensing*

9.2 Compressed Sensing MRI

Compressed sensing has been used [47] to shorten magnetic resonance imaging (MRI) scanning sessions on conventional hardware. Compressed sensing addresses the issue of high scan time by enabling faster acquisition by measuring fewer Fourier coefficients. This produces a high-quality image with relatively lower scan time [11]. The theory prsented in the following chapters is heavily inspired by Wen-mei's and W. Hwu's book, GPU Computing Gems Emerald Edition [48].

9.2.1 Problem Statement

Magnetic resonance imaging suffers from inherent problems that relate to the time that the magnetic resonance images need in order to be acquired. Relevant examinations in clinical contexts are rather timely and uncomfortable for the patient, while they also pose as an expensive procedure. MR imaging is an incredibly flexible imaging modality

with numerous medical applications. The aforementioned disadvantages, however, limit MRI's applicability in many contexts. The reasons that cause these problems are two, the first being the necessity to acquire the MR measurements in a sequential way. Each sample is expensive, since an MRI scanner produces discrete samples of the frequency space, which in clinical contexts is mostly referred to as k-space, one at a time by applying a series of time-varying magnetic fields in the presence of a very strong constant magnetic field. There are unavoidable physical limits on the rate of change of these magnetic fields, as well as the rate at which the molecules in the patient's body can respond to the fields. The second reason for MRI's limitations is its inherently low signal-to-noise ratio (SNR). If too little time is spent acquiring the MRI data, then the image will be dominated by the unavoidable noise generated by the MR system and the patient's body.

This is the point when Compressed Sensing enters the discussion. In fact, compressed sensing is an approach to ameliorate the first cause; A CS MRI scan simply acquires fewer data points.

Recall that Compressed Sensing requires the desired solution to be represented by a sparse vector. Regarding sparsity, a vector itself that is not sparse may be sparse when represented via some other basis. This will occur, for example, when the vector is simply a linear combination of a few of the vectors in the basis. For example a sine wave is not sparse, but has a very compact representation in the Fourier domain. Some vectors may not be sparse, but may be compressible — the magnitudes of their coefficients in some basis decay rapidly. Such a vector is nearly sparse, except for a large number of small coefficients.

In MRI, the objective is to produce an image of the density of tissues on a patient's body. We can denote an image as $x \in C^n$: an n-dimensional complex-valued vector. n is the number of voxels in our 3-D images — typically 100 million or more. Using the MRI scanner we want to take m samples of the k-space (i.e., Fourier domain) representation of our image, with $m \ll n$ to accelerate the scan. We can represent the process of taking these measurements as multiplication of x with a matrix Φ, constructed by selecting some subset of the rows of a discrete Fourier transform (DFT) matrix. We'll denote the undersampled k-space data $y = \Phi x$. Because $m \ll n$, the problem of inverting the equation $y = \Phi x$ is highly underdetermined: a solution can be drawn from an entire $(m - n)$-dimensional subspace of C^n. Compressed sensing provides a means of selecting a particular solution $\tilde{x} \in C^n$ (of the many which satisfy $\Phi x = y$): the image which is maximally sparse. Formally, our desired image can be found by solving the following optimization problem:

$$\text{Minimize} \quad \|Wx\|_1 \tag{9.1}$$

$$\text{Subject to} \quad y = \Phi x \tag{9.2}$$

Where the L_1 norm is defined as the sum of the absolute value of a vector's components: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$; and Wx is the representation of x in a wavelet basis.

9.2.2 Multi-coil Magnetic Resonance Imaging

For multi-coil MRI the above problem is reformulated as follows [12]: the vector of measurements $y_i \in C^L$ acquired by the ith coil can be described by the forward model [13]:

$$y_i = PFS_i x^* + w_i, \; i = 1, \ldots, N_c \tag{9.3}$$

where $x^* \in C^N$ is the image containing N pixels, S_i is an operator representing the point-wise multiplication of the i-th coil sensitivity map, F is the spatial Fourier transform operator, P represents the k-space sampling operator, and we assume $w_i \sim \mathcal{N}_c(0, \sigma^2 I)$ for simplicity. The same under-sampling operator is applied to all N_c coils.

The acceleration factor R denotes the degree of under-sampling in the k-space domain, i.e., $R = N/L$. Due to the multiple coils, the measurements may not be compressive for small R. However, due to redundancy between the coils, the measurements are compressive for moderate values of R [49].

Given multi-coil measurements y, sensitivity maps represented by S and the sampling operator P, the goal of MR image reconstruction is to estimate the underlying image variable x^*. Prior work formulates this as a regularized optimization problem:

$$\arg\min \|y - Ax\|^2 + \lambda \mathcal{Q}(x), \tag{9.4}$$

where we use the operator $A \in C^{M \times N}$ (with $M = N_c \cdot L$) to subsume the discrete approximation to all linear effects, and $\mathcal{Q}$ is a suitably chosen functional prior for the image variable x. For example, to enforce a sparsity prior, one can penalize the L_1 norm in the wavelet representation of x, same as in Eq. 9.1 and 9.2.

9.3 Compressed Sensing MRI with score-based models

Jalal et al. [12] presented an approach to solve inverse problems by employing score-based ideas. Specifically, they presented a general framework to approach linear inverse problems by using posterior sampling via Langevin Dynamics, which can also be applied to perform Compressed Sensing.

Given an observation of the form:

$$y = Ax^* + w, \text{ where } y \in C^{M \times N}, w \sim N_c(0, \sigma^2 I) \text{ and } x^* \sim \mu \tag{9.5}$$

the posterior sampling recovery algorithm outputs x^* according to the posterior distribution $\mu(\cdot|y)$.

Recall the Annealed Langevin Dynamics Algorithm 6.1, which iteratively computes a sample that slowly converges to a realistic sample of the data distribution whose score a neural network (score-based model) is trained on. The ALD method is guaranteed to converge.

So far, it is known that training a model to learn the gradient of the log-likelihood of a data distribution is very much possible. Therefore, one could train such a score-based

model to learn the score of an MRI distribution. But, what about the measurements y? The problem, in fact, requires to utilize the measurements y to reconstruct the desired MRI x^*.

By taking a look at Eq. 9.5 we see that y's distribution given x is the same as its noisy input, i.e. Gaussian. Therefore, we obtain:

$$\nabla_{x_t} \log p(y|x^*) = \frac{A^H(y - Ax_t)}{\sigma^2} \tag{9.6}$$

Ajil et al. [12] also found it useful to anneal this term by setting it to: $\frac{A^H(y-Ax_t)}{\sigma^2+\gamma_t^2}$, where γ_t is a decreasing sequence that ultimately reaches zero. An application of Bayes's rule gives:

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t) + \frac{A^H(y - Ax_t)}{\sigma^2 + \gamma_t^2} \tag{9.7}$$

The above result is remarkable, since it demostrates how an already existing algorithm can be simply enriched with an extra information that ultimately solves a different problem than random generation. It implies that given a network that models the score of a distribution, we can reconstruct samples of this distribution by observing only small parts of them according to a linear schema, Eq. 9.5. The extended Annealed Langevin Dynamics algorithm, which we name Guided Langevin Dynamics, is presented below:

ALGORITHM 9.1: *Guided Langevin Dynamics*

Require: $\{\sigma\}_{i=1}^{L}, \epsilon, T$
1: Initialize $\tilde{x}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad a_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$
4: $\quad$ **for** $i \leftarrow 1$ to T **do**
5: $\quad\quad z_t \sim \mathcal{N}(0, I)$
6: $\quad\quad \tilde{x}_{t+1} \leftarrow \tilde{x}_t + \frac{a_i}{2}\left[s_\theta(\tilde{x}_{t-1}, \sigma_i) + \frac{A^H(y-A\tilde{x}_{t-1})}{\sigma^2+\gamma_t^2}\right] + \sqrt{a_i}\, z_t$
7: $\quad \tilde{x}_0 \leftarrow \tilde{x}_T$
return $\tilde{x}_0$

The above result is employed to also perform Compressed Sensing, simply by setting the measurements matrix A according to Eq. 9.3.

Part III

Contribution and Experiments

Chapter 10

Bijective Relation between score-based and diffusion models

Score-based models and diffusion models are both trained by perturbing the data with specific noise schedules. As presented in previous chapters, the noise schedules are either discrete and prescribed for each model, or continuous and described by Stochastic Differential Equations. Based on which method is used during training, each model produces a certain quantity at its output which is meant for solving the reverse procedure of the one used for training. For instance, the score function at the output of an NCSN is meant for use in a VE SDE solver. In this chapter, there is established a bijective relation between the two main SDEs (VE and VP) and it is shown how a model that is trained on one equation can be used to solve the other.

10.1 Equivalence between Variance Exploding and Variance Preserving diffusion models

The following proof is inspired by the proof given in [14] Appendix B.

Proposition 10.3. *The Variance Exploding and Variance Preserving specifications can be considered equivalent up to rescaling of the noisy latents x_t.*

Proof. For the Variance Exploding SDE, the noise schedule is: $\sigma_t \in [0, +\infty)$ and for the Variance Preserving SDE, the noise schedule is: $a_t \in (0, 1]$ (we use the a_t notation, which was so far represented as $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_t)$). The corresponding intermediate distributions of each process are: $q(x_t^{VE}|x_0) = \mathcal{N}(x_t^{VE}; x_0, \sigma_t^2 I)$ and $q(x_t^{VP}|x_0) = \mathcal{N}(x_t^{VE}; \sqrt{a_t}x_0, (1-a_t)I)$. By making use of Lemma 7.1 we can express a perturbed datum at timestep t as follows:

$$x_t^{VE} = x_0 + \sigma_t z, \quad z \sim \mathcal{N}(0, I) \tag{10.1}$$

$$x_t^{VP} = \sqrt{a_t}\, x_0 + \sqrt{1-a_t}\, z, \quad z \sim \mathcal{N}(0, I) \tag{10.2}$$

We divide each member of Eq. 10.2 by $\sqrt{a_t}$ and we have:

$$\frac{x_t^{VP}}{\sqrt{a_t}} = x_0 + \frac{\sqrt{1-a_t}}{\sqrt{a_t}} z = x_0 + \sqrt{\frac{1}{a_t} - 1}\, z$$

We see that $\sqrt{\frac{1}{a_t} - 1}$ ranges from 0 (inclusive) to $+\infty$ (exclusive) in a reverse fashion that σ_t does.

It is obvious that if we let $\sqrt{\frac{1}{a_t} - 1} = \sigma_t$ (which can naturally be done due to the above observation) then the quantities x_t^{VE} and $\frac{x_T^{VP}}{\sqrt{a_t}}$ are equal. Therefore, we see the following bijections between the two SDE specifications:

$$x_t^{VE} \longleftrightarrow \frac{x_t^{VP}}{\sqrt{a_t}} \tag{10.3}$$

$$\sigma_t \longleftrightarrow \sqrt{\frac{1}{a_t} - 1} \tag{10.4}$$

$$\frac{1}{1 + \sigma_t^2} \longleftrightarrow a_t \tag{10.5}$$

Based on the above, we see that we can incorporate a diffusion model trained on one SDE into a sampler that solves the other. □

10.2 Relation between score-based and diffusion model outputs

A score-based model is trained to output the score function $s : \mathbb{R}^d \to \mathbb{R}^d$ of its input x. On the other hand, a diffusion model is trained to produce the noise $\epsilon \sim \mathcal{N}(0, I)$ that perturbs its input x. Between these two outputs, there is an implicit relation.

Recall that $s_\partial(x) \approx \nabla_x \log p(x)$ and the distribution $p(x)$ is Gaussian. Thus:

$$s_\partial(x) \approx \nabla_x \log p(x) = \nabla_x \Big(-\frac{1}{2\sigma^2}(x - \mu)^2 \Big) = -\frac{x - \mu}{\sigma^2} = -\frac{\epsilon}{\sigma}$$

We observe that a score-based model's output is also the noise $\epsilon \sim \mathcal{N}(0, I)$ that perturbs its input x divided by the noise's standard deviation.

Similarly, considering VP SDE perturbations, i.e. perturbation kernels $q(x_t|x_0) \sim \mathcal{N}(\sqrt{\bar{a}_t}x_0, (1 - \bar{a}_t)I)$, we have:

$$s_\partial(x_t, t) \approx \nabla_{x_t} \log q(x_t) = \mathbb{E}_{q(x_0)}[\nabla_{x_t} q(x_t|x_0)] = \mathbb{E}_{q(x_0)}\Big[-\frac{\epsilon_\partial(x_t, t)}{\sqrt{1 - \bar{a}_t}} \Big] = -\frac{\epsilon_\partial(x_t, t)}{\sqrt{1 - \bar{a}_t}} \tag{10.6}$$

10.3 Transforming DDIM into Score-based Implicit Model (SBIM)

From the previous chapters of Part III, we can deduce that diffusion samplers can be used by either a score-based or a diffusion model. In Chapter 4, we presented the DDIM algorithm (7.3) which is performed by using a diffusion model (VP specification) and therefore solves the reverse Variance Preserving SDE. In this chapter, we transform the DDIM sampling algorithm into score-based DDIM, i.e. we reparameterize it to employ a score-based model to solve the VP SDE.

Recall the main iterative rule of DDIM:

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, x_{0t}^{VP} + \sqrt{1-a_{t-1}}\, \epsilon_\partial(x_t^{VP}) \overset{\text{Lemma } 7.1}{\iff}$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} - \sqrt{1-a_t}\, \epsilon_\partial(x_t^{VP})}{\sqrt{a_t}} + \sqrt{1-a_{t-1}}\, \epsilon_\partial(x_t^{VP}) \overset{\text{Eq. } 10.6}{\iff}$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} + \sqrt{1-a_t} \cdot \sqrt{1-a_t}\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}\, s_\partial(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\, \frac{x_t^{VP} + (1-a_t)\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}\, s_\partial(x_t^{VE}, t)$$

Given that our score-based model is trained on a Variance Exploding noise schedule, its input must be accordingly perturbed. Therefore, every time we input x_t into our model, we must first divide it by $\sqrt{a_t}$ as by Eq. 10.3. We can divide both members of the last iteration rule by a_{t-1} **without changing the Variance Preserving property**. At the same time, we know that x^{VE} and x^{VP} converge to the same x_0, since a_t is slowly decreasing. Based on the above and Eq. 10.3 we get:

$$x_{t-1}^{VE} \leftarrow \frac{\sqrt{a_t}\, x_t^{VE} + (1-a_t)\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\, s_\partial(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \frac{(1-a_t)\, s_\partial(x_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\, s_\partial(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{1-a_t}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\right] s_\partial(x_t^{VE}, t)$$

From Eq. 10.5, we have:

$$1 - a_t = 1 - \frac{1}{1+\sigma_t^2} = \frac{\sigma_t^2}{1+\sigma_t^2} \tag{10.7}$$

From Eq. 10.7 and 10.5, we have:

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{\frac{\sigma_t^2}{1+\sigma_t^2}}{\sqrt{\frac{1}{1+\sigma_t^2}}} - \frac{\sqrt{\frac{\sigma_{t-1}^2}{1+\sigma_{t-1}^2}} \cdot \sqrt{\frac{\sigma_t^2}{1+\sigma_t^2}}}{\sqrt{\frac{1}{1+\sigma_{t-1}^2}}}\right] s_\partial(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \left[\frac{\sigma_t^2}{\sqrt{1+\sigma_t^2}} - \frac{\sigma_{t-1} \cdot \sigma_t^2}{\sqrt{1+\sigma_t^2}}\right] s_\partial(x_t^{VE}, t) \iff$$

$$x_{t-1}^{VE} \leftarrow x_t^{VE} + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\Big[\sigma_t - \sigma_{t-1}\Big]\, s_\partial(x_t^{VE}, t)$$

Therefore, we can use a score-based model to perform DDIM, which has VP specifica-

tions, according to the following algorithm:

ALGORITHM 10.1: *Score-based Implicit Model*

Require: $\{\sigma_t\}_{t=0}^{N}$
1: $x_T \sim \mathcal{N}(0, I)$
2: **for** $t \leftarrow T$ to 1 **do**
3: $\quad x_{t-1} \leftarrow x_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}} \left(\sigma_t - \sigma_{t-1}\right) \cdot s_\theta(x_t, t)$
4: **return** x_0

The meaning of the above derivation lies on the fact that DDIM has been empirically observed to produce great samples when performed on a subset of the original steps that the model in use was trained on. Therefore, it would be of great interest to test this when using a score-based model.

Chapter 11

Solving linear inverse problems via Score-based Implicit Model

In chapter 9, we presented a general framework to solve linear inverse problems with the use of posterior sampling via Annealed Langevin Dynamics. Jalal et al. [12] showed that by observing measurements of the form $y = Ax^* + w$, where $w \sim N_c(0, \sigma^2 I)$, we can reconstruct the signal x^* if we have access to the score function of x^*'s distribution by applying Guided Langevin Dynamics (9.1). However, there is a limitation from which Langevin Dynamics inherently suffer, and that is the sampling speed. Langevin Dynamics needs thousands of sampling steps, usually iterating over the same noise level multiple times in order to produce good results. In this chapter we extend score-based implicit model to solve linear inverse problems of the form of Eq. 9.5.

11.1 Guided score-based implicit model

Inspired by Ajil et al. [12], we extend Algorithm 10.1 to incorporate the measurements information in the iterative process. The same math apply for this case, as they are described in Chapter 9. In order to use the measurements we want to sample from the distribution $p(x|y)$, which corresponds to a score function of the form of Eq. 9.7. Recall:

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \frac{A^H(y - Ax)}{\sigma^2 + \gamma_t^2}$$

Following the same thinking, we add the guiding quantity in our score-based implicit model algorithm 10.1 to perform *Guided score-based implicit model*.

11.2 One extra denoising step

Jolicoeur-Martineau et al. [50] noticed that adding an extra denoising step after the original annealed Langevin dynamics 6.1, often significantly improves FID scores without affecting the visual appearance of samples. Instead of directly returning x_0, this denoising step returns $x_0 + \sigma_0^2 s_\theta(x_0, \sigma_0)$, which essentially removes the unwanted noise $\mathcal{N}(0, \sigma_0^2 I)$ from x_0 using Tweedie's formula [15].

The final algorithm after adding this last denoising step is presented below:

ALGORITHM 11.1: *Guided score-based Implicit Model*

Require: $\{\sigma_t\}_{t=0}^{N}$
1: $x_T \sim \mathcal{N}(0, I)$
2: **for** $t \leftarrow T$ to 1 **do**
3: $\quad x_{t-1} \leftarrow x_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\left(\sigma_t - \sigma_{t-1}\right) \cdot \left(s_\partial(x_t, t) + \frac{A^H(y - Ax_t)}{\sigma^2 + \gamma_t^2}\right)$
4: **return** $x_0 + \sigma_0^2\, s_\partial(x_0, \sigma_0)$

11.3 Guided score-based implicit model with Langevin Dynamics corrector

Inspired by [3]'s proposition of predictor-corrector samplers that solve SDE's, we propose to mix Guided score-based implicit model, which is going to play the role of the predictor, with Langevin Dynamics, which is meant to be the corrector. This sampler will no longer be deterministic, however.

ALGORITHM 11.2: *Guided score-based implicit model with LD corrector*

Require: $\{\sigma_t\}_{t=0}^{N}$, M
1: $x_T \sim \mathcal{N}(0, I)$
2: **for** $t \leftarrow T$ to 1 **do**
3: $\quad x_{t-1} \leftarrow x_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\left(\sigma_t - \sigma_{t-1}\right) \cdot \left(s_\partial(x_t, t) + \frac{A^H(y - Ax_t)}{\sigma^2 + \gamma_t^2}\right)$
4: $\quad a_i \leftarrow \epsilon \cdot \sigma_t^2 / \sigma_L^2$
5: $\quad$ **for** $i \leftarrow 1$ to M **do**
6: $\quad\quad z \sim \mathcal{N}(0, I)$
7: $\quad\quad x_{t-1} \leftarrow x_{t-1} + \frac{a_i}{2}\left(s_\partial(x_{t-1}, t-1) + \frac{A^H(y - Ax_{t-1})}{\sigma^2 + \gamma_{t-1}^2}\right) + \sqrt{a_i}\, z$
8: **return** $x_0 + \sigma_0^2\, s_\partial(x_0, \sigma_0)$

Chapter 12

Compressed Sensing MRI via guided score-based implicit model

After establishing our framework to solve linear inverse problems using guided score-based implicit model, it is of great interest to us to investigate how these new samplers can be used to accelerate Compressed Sensing MRI. Our code that replicates the experiments and given plots can be found on: https://github.com/d-dimos/thesis_ntua_sbim.

12.1 NYU fastMRI Dataset

For our experiments, we use a score-based model, NCSNv2 [30], that is pretrained on a small subset of scans from the NYU fastMRI brain dataset. Specifically, Ajil et al. [12] trained it using T2-weighted images at a field strength of 3 Tesla for a total of 14,539 2D training slices. They applied the ESPIRiT algorithm to the fully-sampled central portion of k-space to estimate the sensitivity maps and calculate the MVUE from the fully sampled data.e to estimate the sensitivity maps.

The fastMRI dataset [51] is a deidentified imaging dataset provided by NYU Langone and comprises raw k-space data in several sub-dataset groups. From fastMRI we use a subset of the T2-weighted brain images to evaluate and compare our algorithms.

12.2 Model Architecture

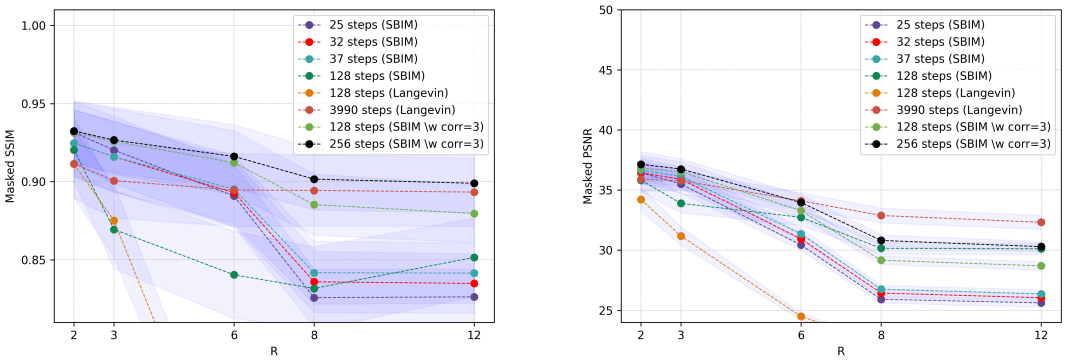
The backbone network for the pretrained model is a RefineNet [52]. The model used is publicly available on [12]'s official repository (https://github.com/utcsilab/csgm-mri-langevin), and is based on the official implementation of [30] (https://github.com/ermongroup/ncsnv2).

12.3 Experimental Results

12.3.1 In-distribution (brain) - Vertical Masking

For our experiements, we employed Jalal et al.'s [12] NCSN pretrained on brain MVUEs model and performed SBIM, SBIM with Langevin corrector, and the original Langevin Dynamics, to perform compressed sensing and evaluate each sampler on three axes; SSIM, PSNR and NFEs. For the experiments, we used a subset of 200 brain MRIs.

The results are presented in Fig. 12.1, and Tables 12.2 and 12.1.



(a) *Masked SSIM* (b) *Masked PSNR*

Figure 12.1: *Sampler Comparison - Vertical Mask*

In the above figures, we present the average test SSIM and PSNR with vertical masking across a range of acceleration factors R. Shaded regions indicate 95% confidence intervals. The mask zeros out all coordinates whose absolute value is smaller than 0.05 times the maximum absolute value in the fully-sampled MVUE, and this reduces the effect of noise in the background. In addition, the mask zeros out all pixels out of a 165-pixel-radius circle. Higher R indicates a smaller number of acquired measurements.

The SBIM algorithm with Langevin corrector produces the best results regarding the SSIM metric accross all R, while it requires approximately 15.5 times less NFEs. When the R factor is lower or equal to 6, the 32-step SBIM seems to also produce better results than Langevin Dynamics, and is approximately 125 times faster.

Regarding the PSNR metric, our best PC-SBIM seems to produce the most noise-free samples for most R values, however Langevin Dynamics still maintains the best PSNR results for $R = 6, 8, 12$, most probably because its denoising procedure has much more time to incorporate the measurements and remove the noise.

R/NFEs	32 (SBIM)	128 (SBIM)	128 (SBIM-3)	128 (LD)	3990 (LD)	256 (SBIM-3)
2	0.92	0.92	**0.93**	0.91	0.91	**0.93**
3	0.92	0.87	**0.93**	0.88	0.90	**0.93**
6	0.89	0.84	0.91	0.68	0.89	**0.92**
8	0.84	0.83	0.89	0.56	0.89	**0.90**
12	0.83	0.85	0.88	0.58	0.89	**0.90**

Table 12.1: *Sampler Comparison - SSIM - Vertical Mask*

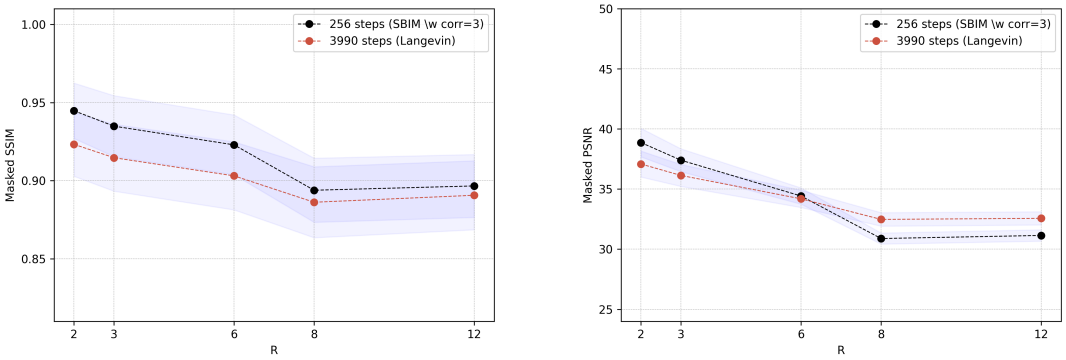
R/NFEs	32 (SBIM)	128 (SBIM)	128 (SBIM-3)	128 (LD)	3990 (LD)	256 (SBIM-3)
2	36.4	35.8	36.8	34.2	35.9	**37.1**
3	35.9	33.9	36.5	31.2	35.8	**36.8**
6	31.0	32.7	33.3	24.5	**34.1**	34.0
8	26.4	30.1	29.2	22.2	**32.9**	30.8
12	26.0	30.1	28.7	23.0	**32.3**	30.3

Table 12.2: *Sampler Comparison - PSNR - Vertical Mask*

On both metrics, SBIM (with various configurations) seems to produce very good results. Langevin Dynamics is significantly slower than SBIM, while the former beats the latter in terms of PSNR only for large acceleration factors, and only slightly so. However, SBIM is always much faster than Langevin Dynamics.

12.3.2 In-distribution (brain) - Horizontal Masking

By repeating the above experiments for the best two samplers, i.e. 256-step SBIM with 3-steps corrector and Langevin Dynamics, using horizontal masking we get very similar results. The configurations regarding the masking and the way SSIM and PSNR are calculated are kept the same, with the only change being the horizontal masking of the MRI. The results are presented in plot form in Fig. 12.2 and in table form in Table 12.3.



(a) *Masked SSIM* (b) *Masked PSNR*

Figure 12.2: *Sampler Comparison - Horizontal Mask*

R/NFEs	256 (SBIM-3)	3990 (LD)
2	**0.94**	0.92
3	**0.93**	0.91
6	**0.92**	0.90
8	**0.89**	0.87
12	**0.90**	0.89

(a) *Masked SSIM*

R/NFEs	256 (SBIM-3)	3990 (LD)
2	**38.9**	37.1
3	**37.4**	36.1
6	**34.4**	34.2
8	30.9	**32.5**
12	31.14	**32.6**

(b) *Masked PSNR*

Table 12.3: *Sampler Comparison - Horizontal Mask*

Chapter 13

Discussion

13.1 Related Work

Other work on fast sampling adjust the generating samplers or their corresponding diffusion processes in manual or automated ways in order to accelerate sample generation. Watson et al. [53] aim at reducing the timesteps needed for a diffusion models to produce samples by proposing a dynamic programming algorithm that optimizes log likelihood. Kong & Ping [54]; Nichol & Dhariwal [43] describe methods to adjust a discrete time diffusion model trained on many timesteps into models that can sample in few timesteps. San-Roman et al. [55] speed up generation by training a new network that estimates the noise level of noisy data and show how to utilize it for acceleration. Salimans & Ho [56] train a student diffusion network to produce the output of a teacher model's two steps in one (distill the teacher), and repeat this procedure to reduce sampling time. Saharia et al. [57]; Chen et al. [35]; Ho et al. [58] use continuous noise scales to train diffusion models and reform the samplers after training in order to perform a fewer-step discrete reverse diffusion.

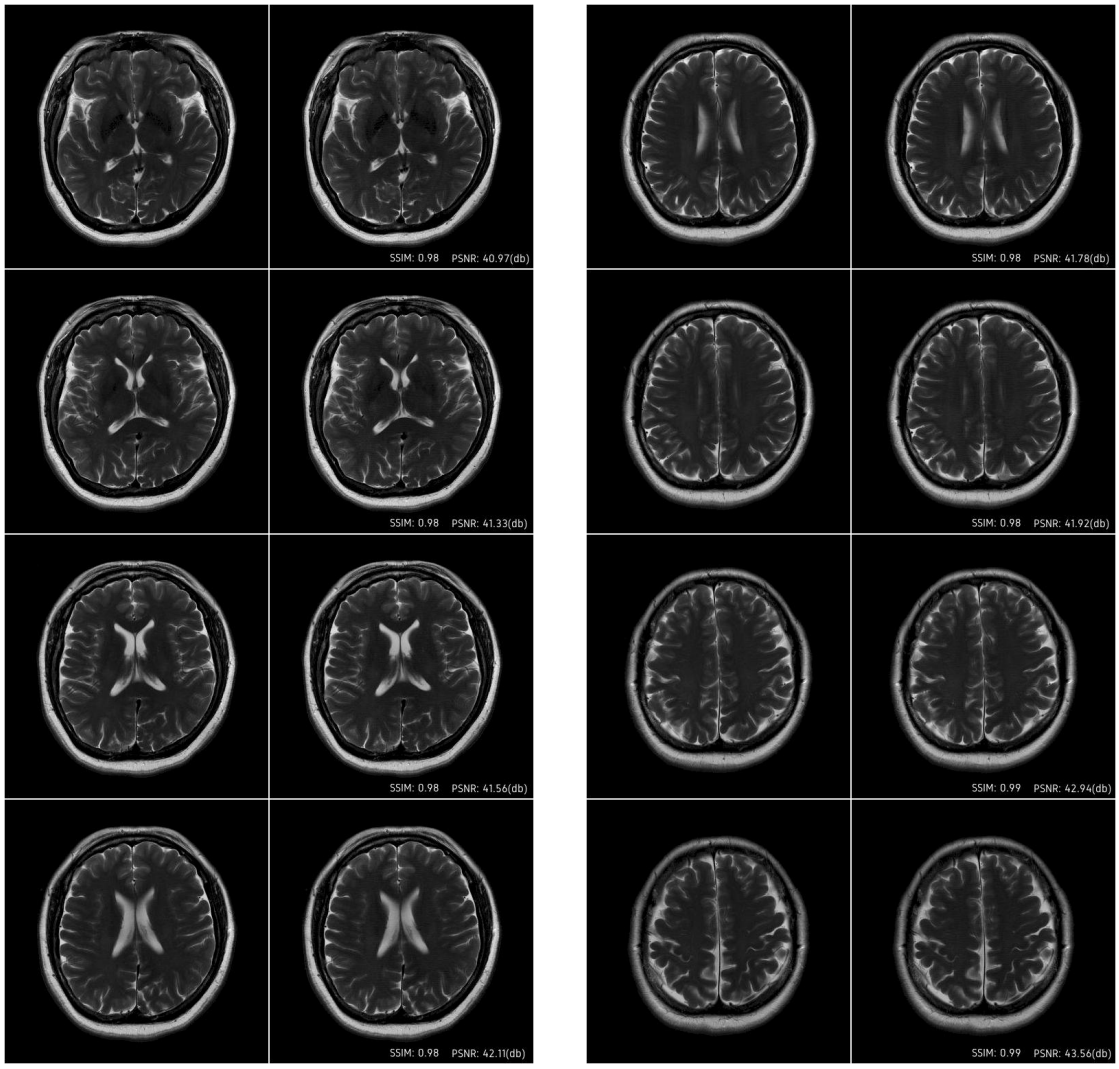
13.2 Conclusion

We presented Score-based Implicit Model (SBIM) and a Predictor-Corrector variation of it. SBIM is a reparameterization of diffusion models' DDIM sampler in a way that allows the latter to employ score-based models to accelerate sample generation. These new algorithms can be used to solve inverse problems, the same way previous Langevin Dynamics frameworks were designed to, in a straightforward fashion. By applying SBIM to perform Compressed Sensing in MRIs, the algorithm is shown to be very competitive in terms of SSIM and PSNR metrics, while being significantly cheaper to run at test time. By presenting this algorithm, we hope to contribute to increasing the usefulness of score-based solutions to inverse problems and practical applications.

Appendices

Appendix A

Score-based Implicit Model Sample Sample Reconstructions



(a) (b)

Figure A.1: *R=2, 32 steps (part 1)*

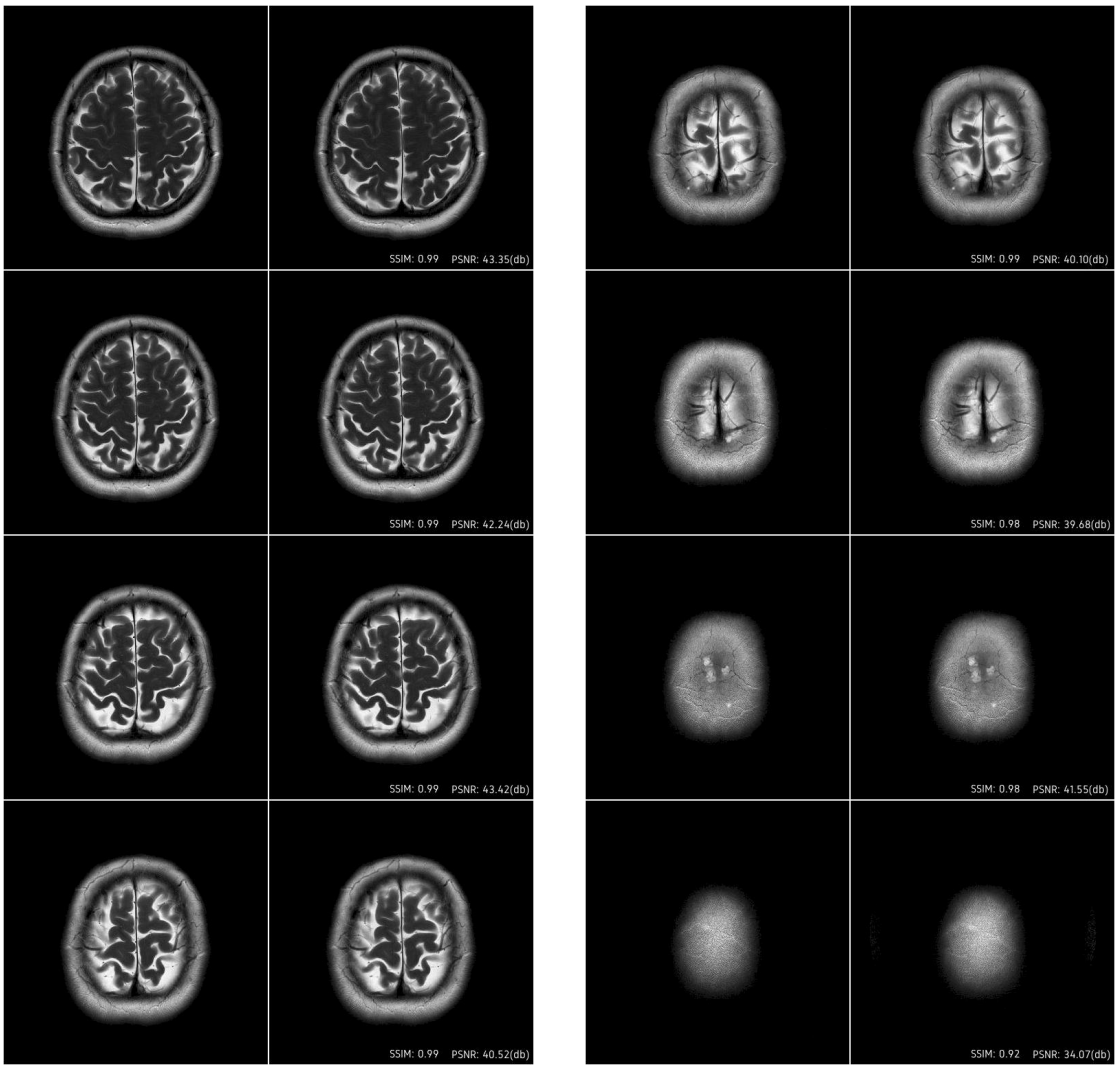
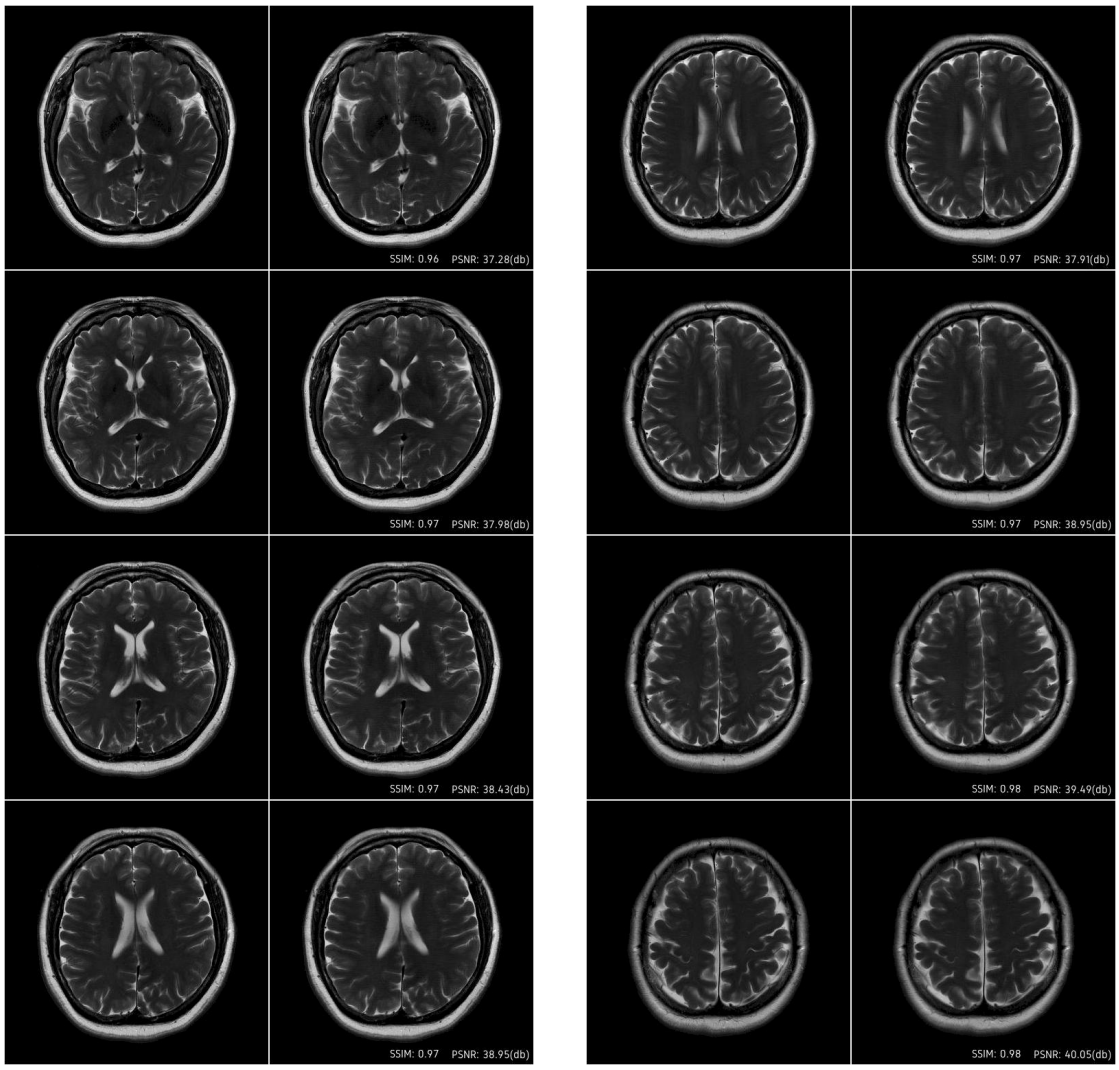
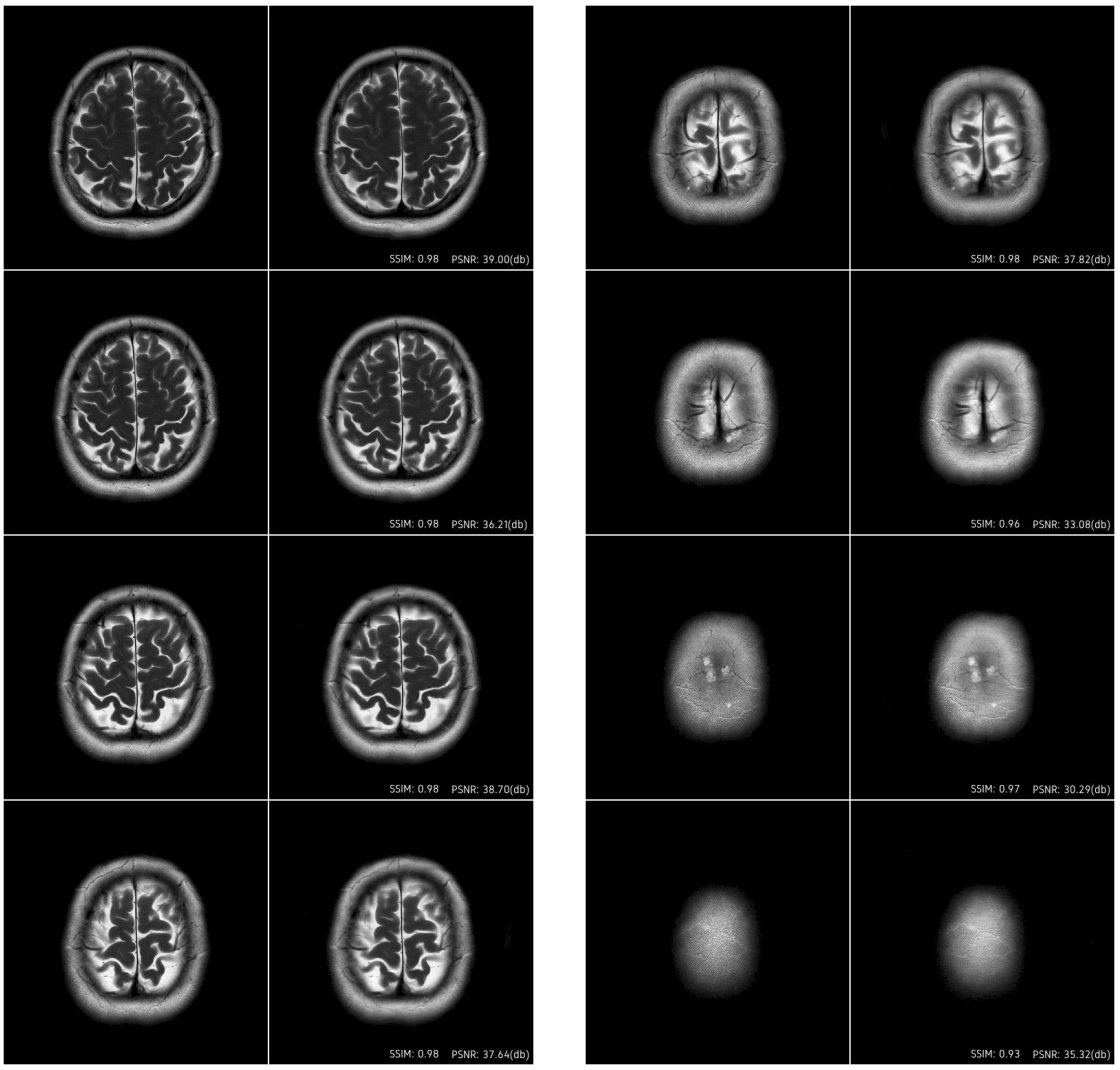


Figure A.2: *R=2, 32 steps (part 2)*



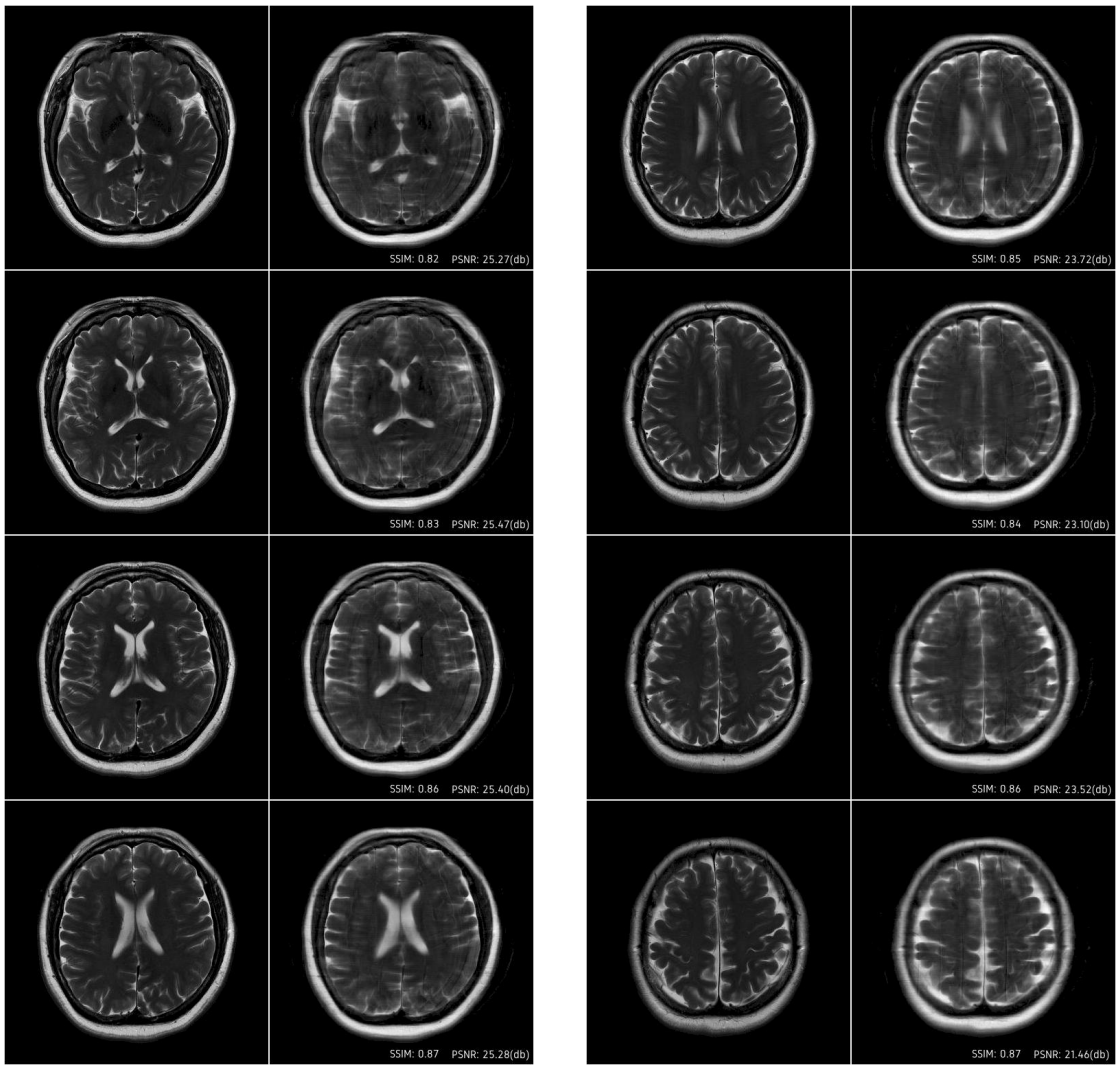
(a) (b)

Figure A.3: *R=4, 32 steps (part 1)*



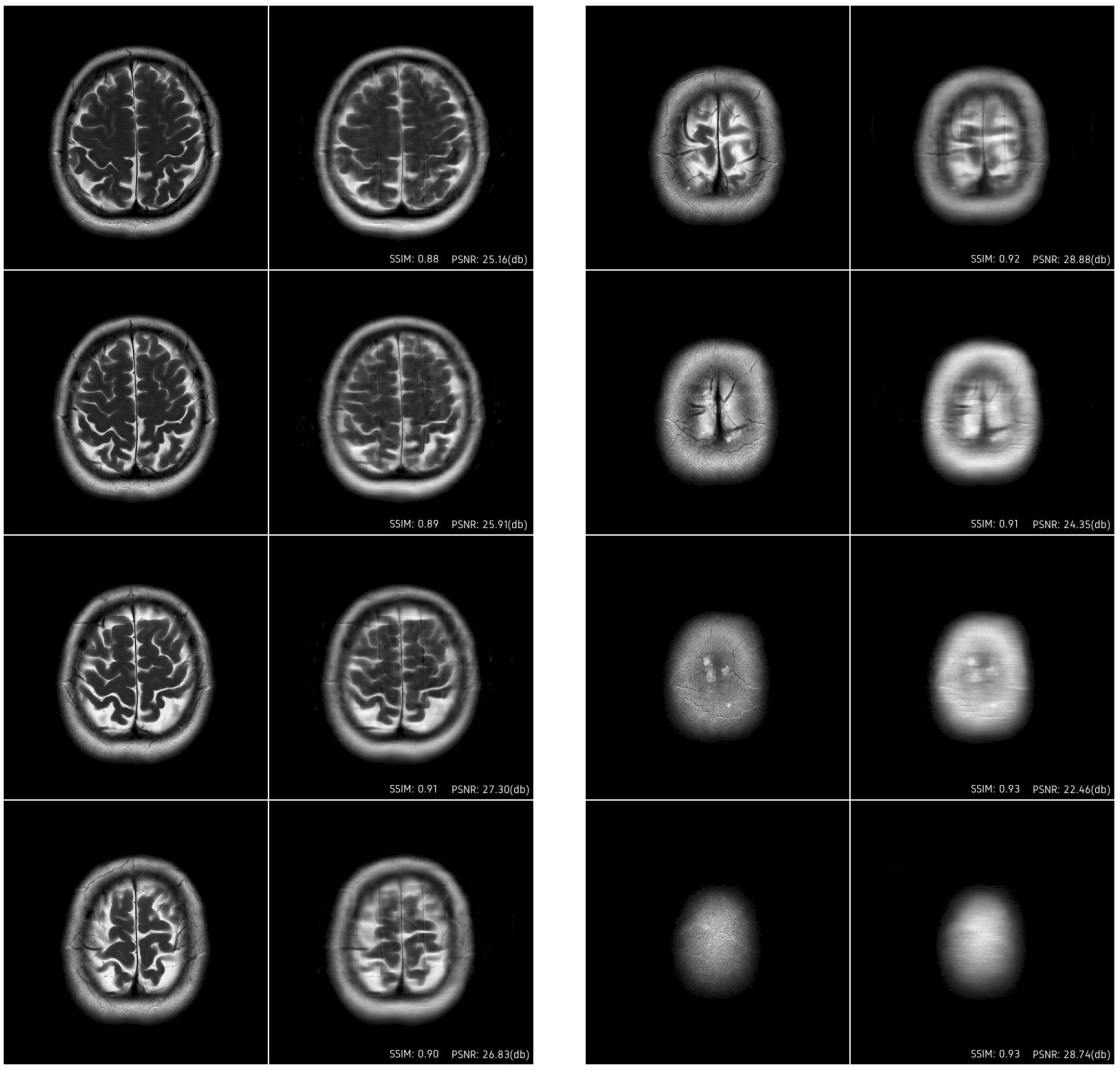
(a) (b)

Figure A.4: *R=4, 32 steps (part 2)*



(a) (b)

Figure A.5: *R=12, 32 steps (part 1)*

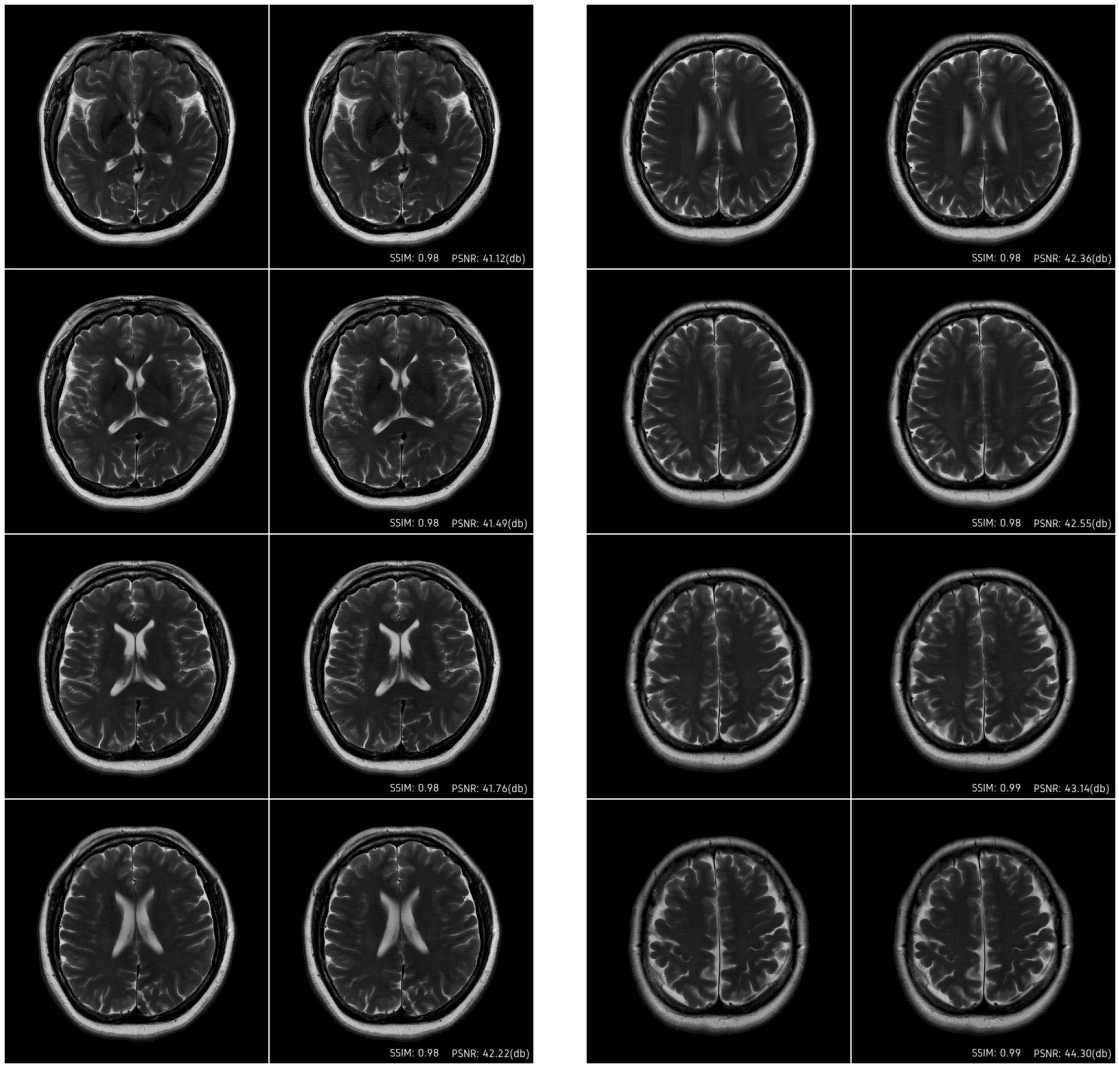


(a) (b)

Figure A.6: *R=12, 32 steps (part 2)*

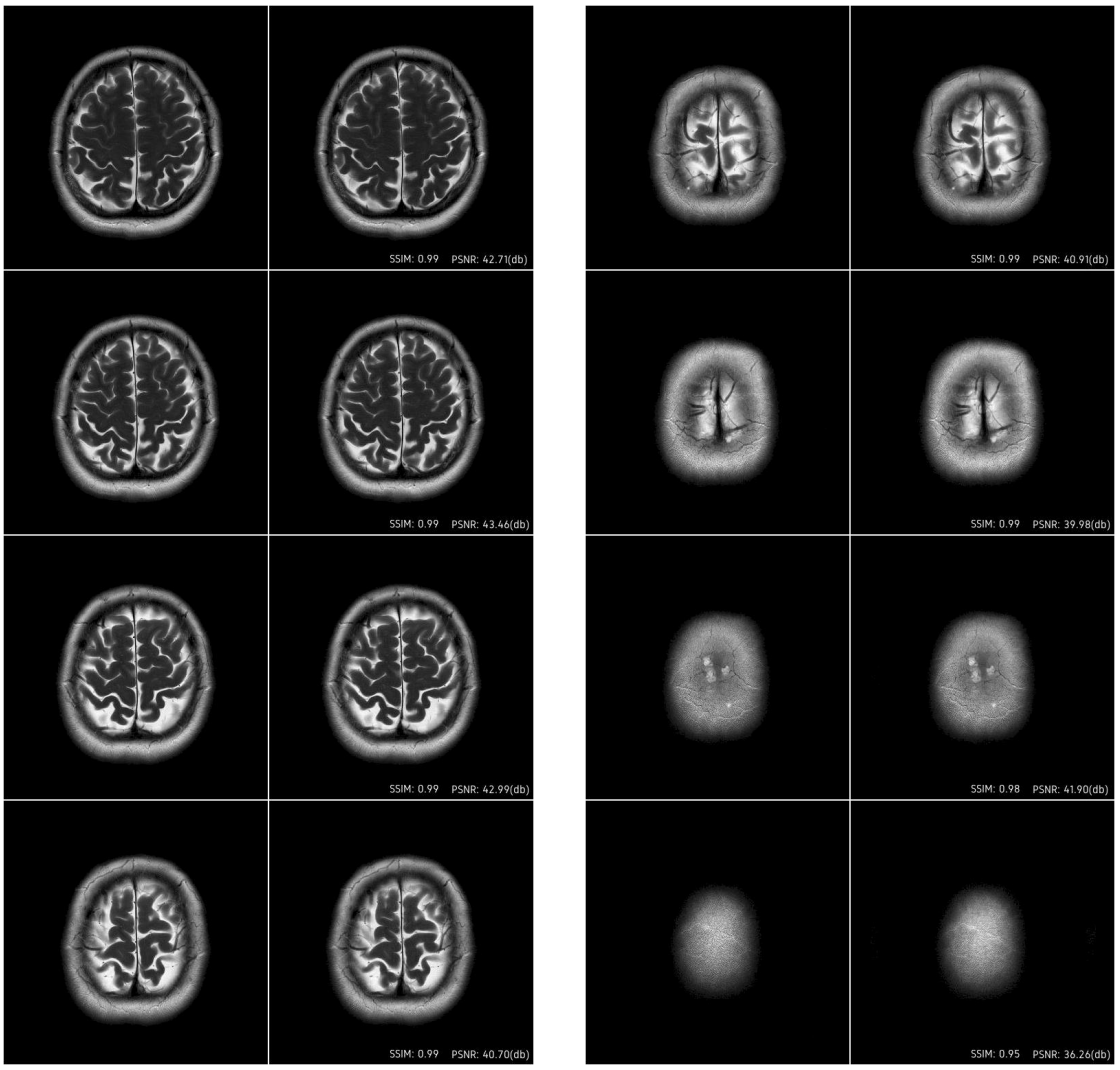
Appendix B

Guided score-based implicit model with LD corrector Sample Reconstructions



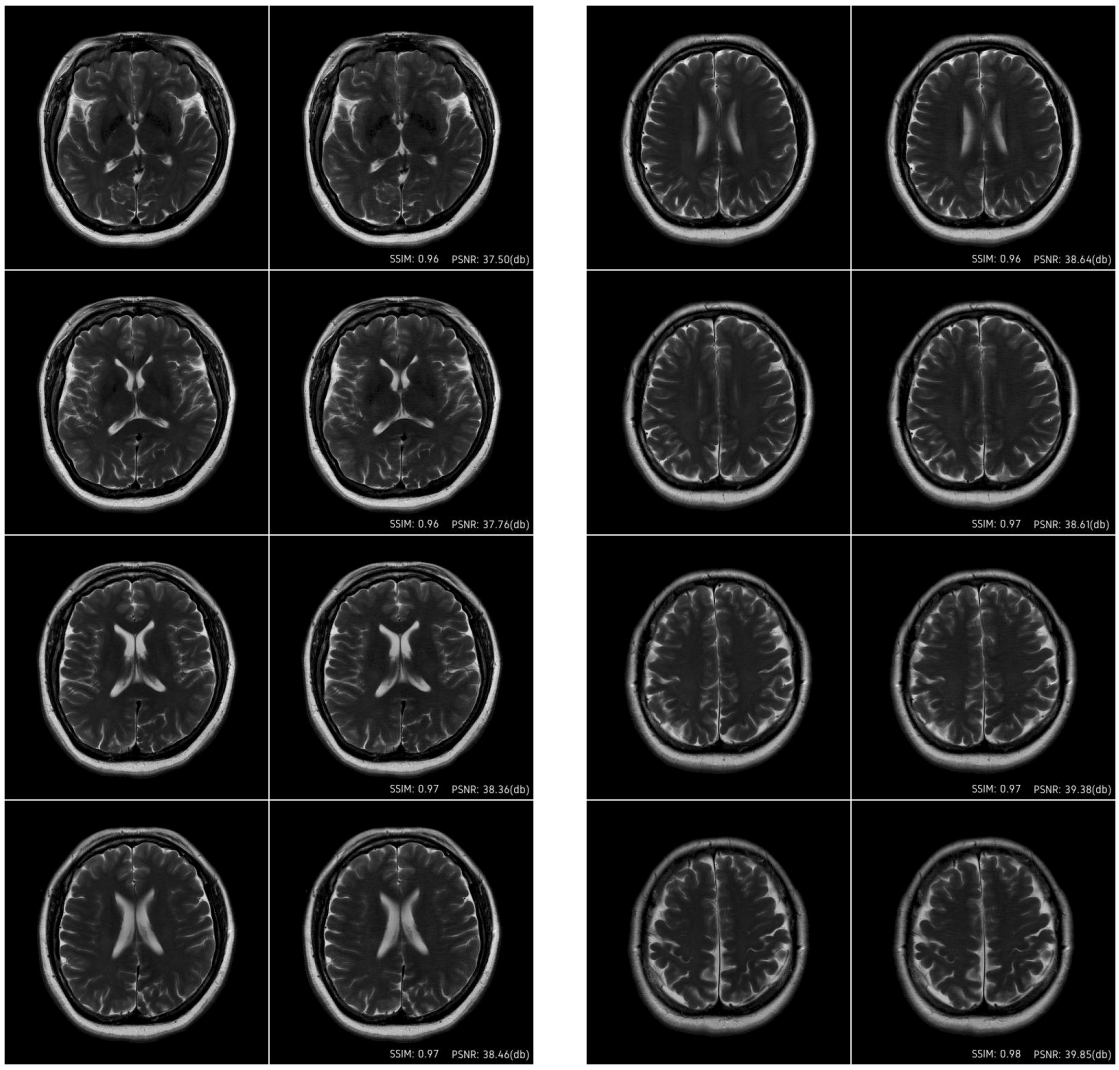
(a) (b)

Figure B.1: *R=2, 32 predictor steps + 64 corrector steps (part 1)*



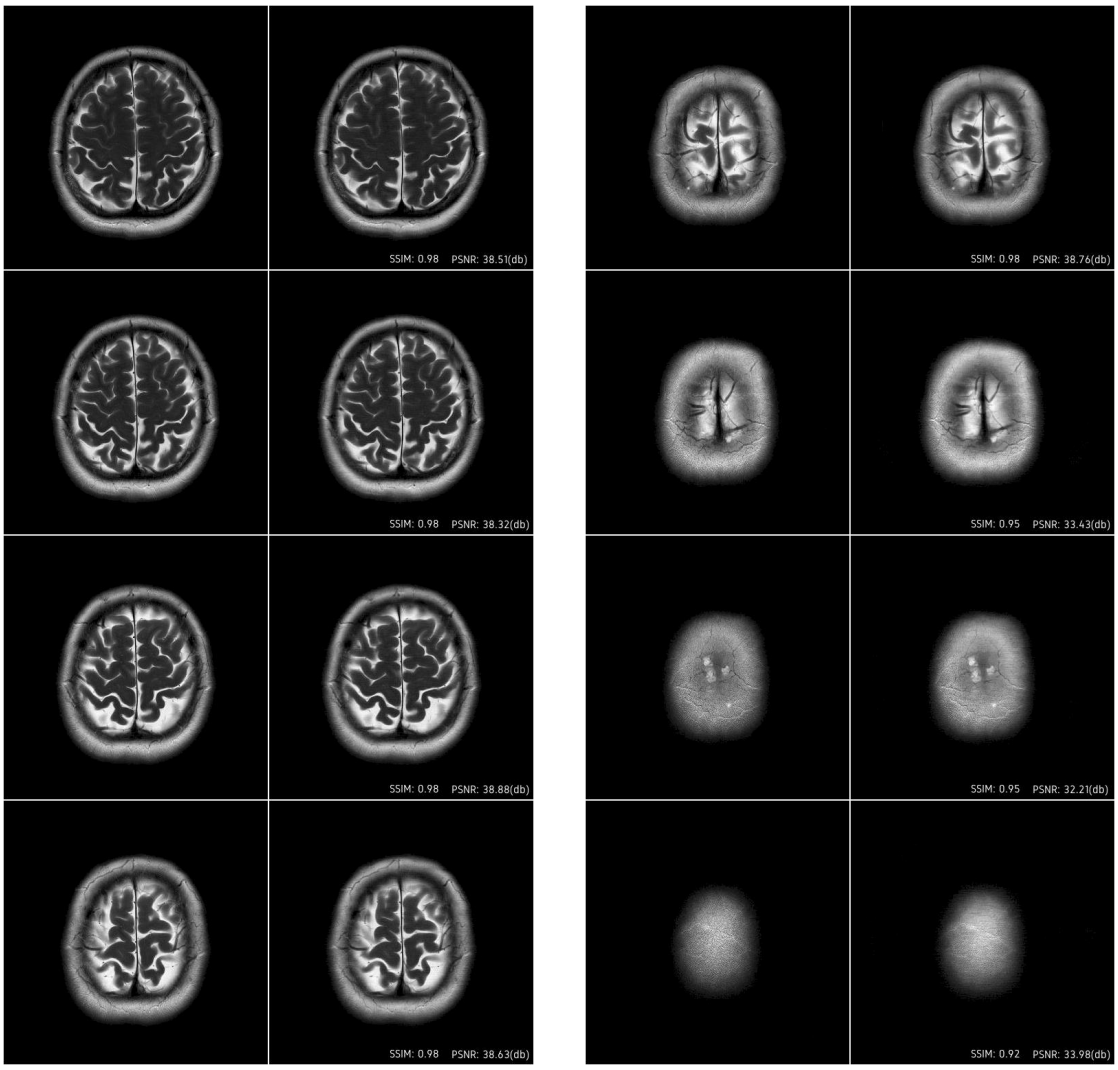
(a) (b)

Figure B.2: *R=2, 32 predictor steps + 64 corrector steps (part 2)*



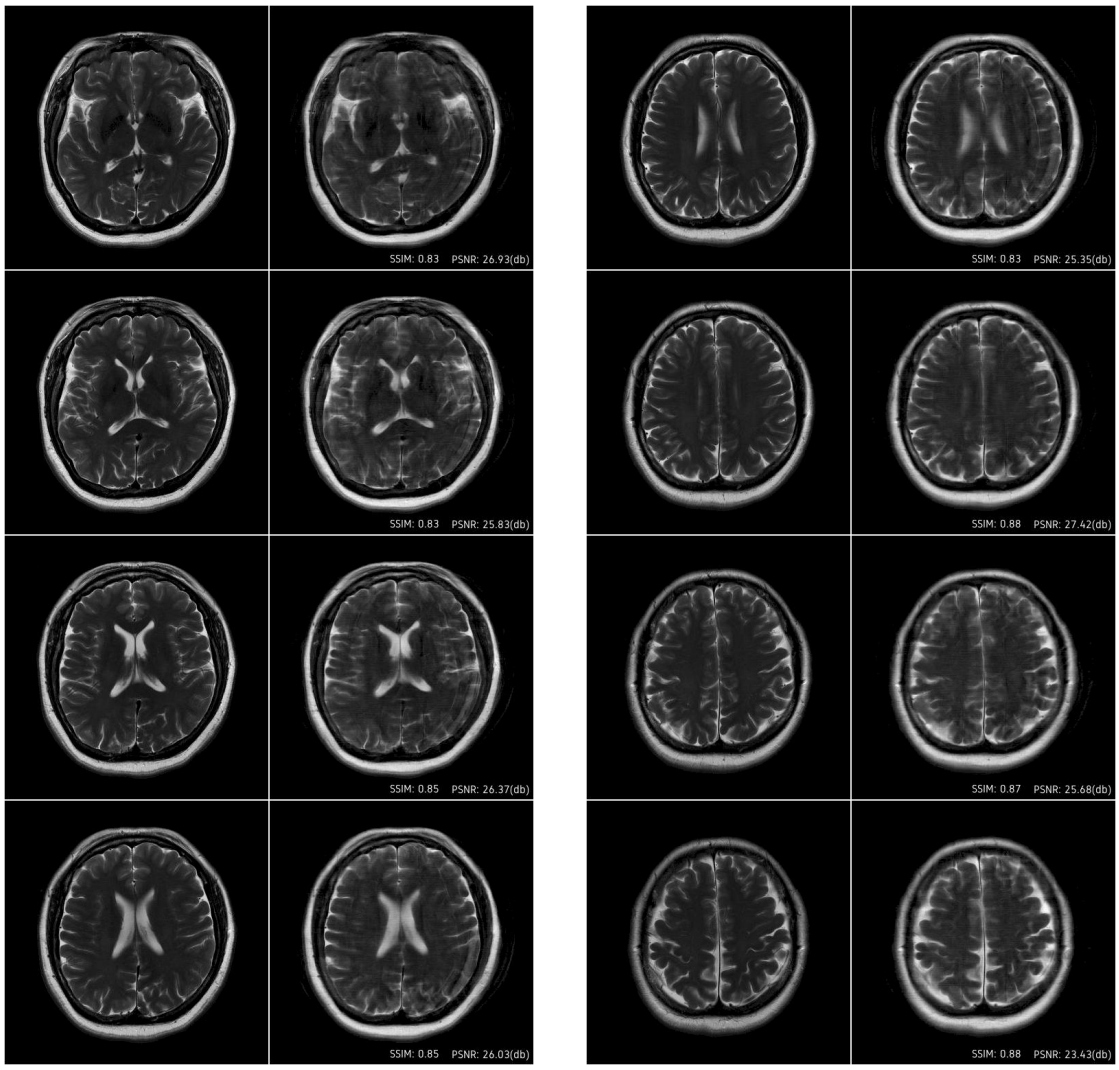
(a) (b)

Figure B.3: *R=4, 32 predictor steps + 64 corrector steps (part 1)*



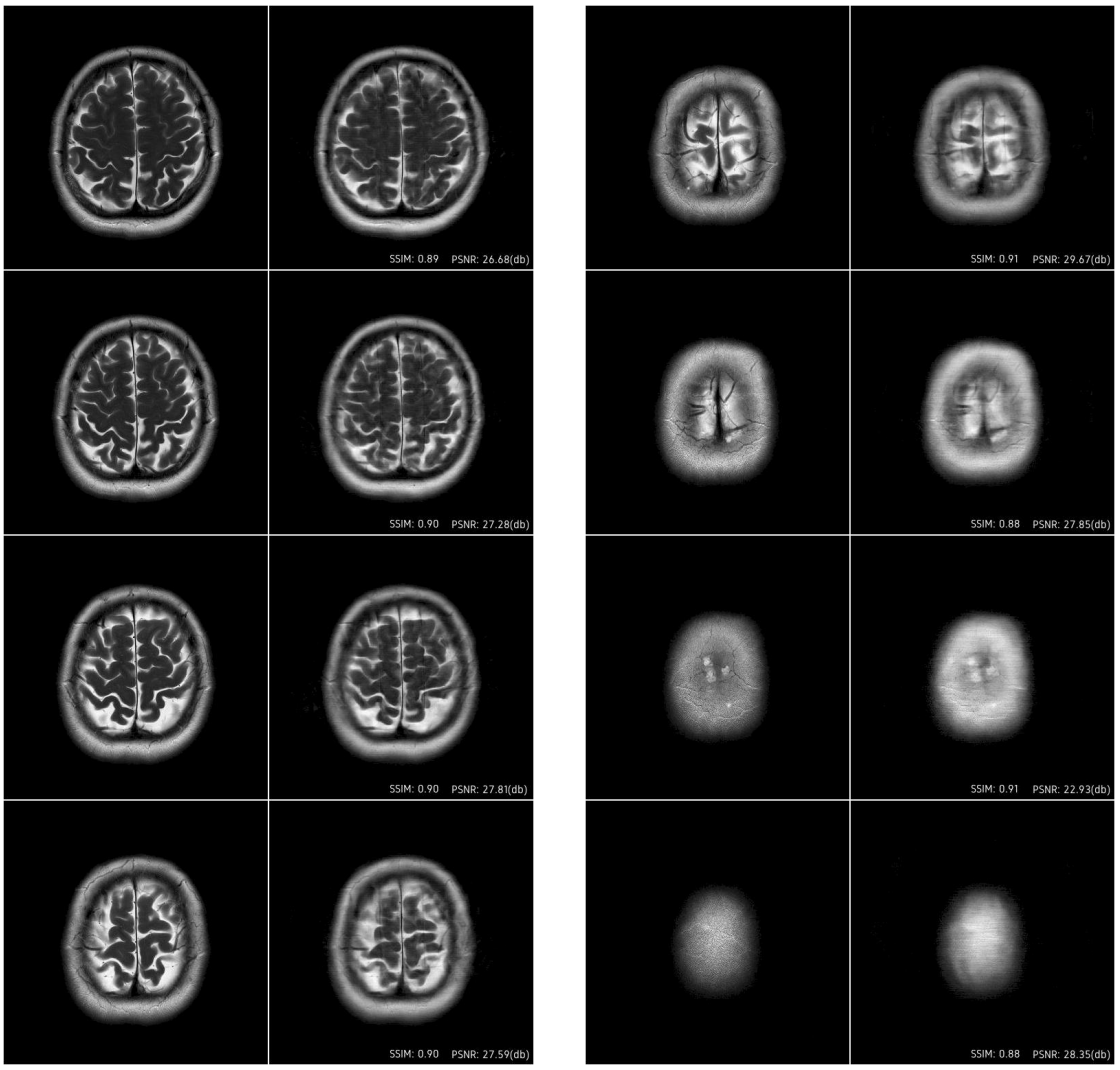
(a) (b)

Figure B.4: *R=4, 32 predictor steps + 64 corrector steps (part 2)*



(a) (b)

Figure B.5: *R=12, 32 predictor steps + 64 corrector steps (part 1)*

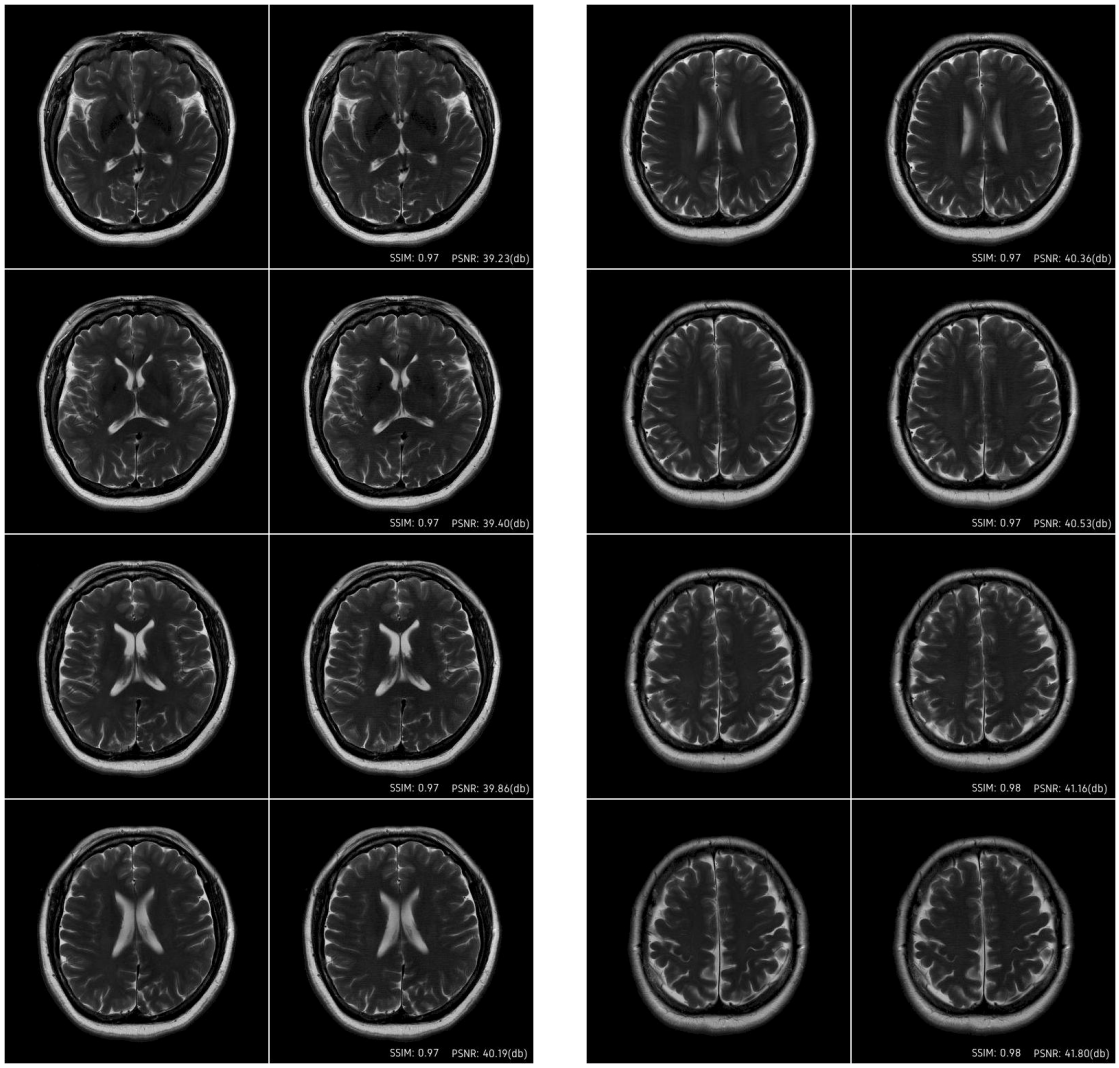


(a) (b)

Figure B.6: *R=12, 32 predictor steps + 64 corrector steps (part 2)*

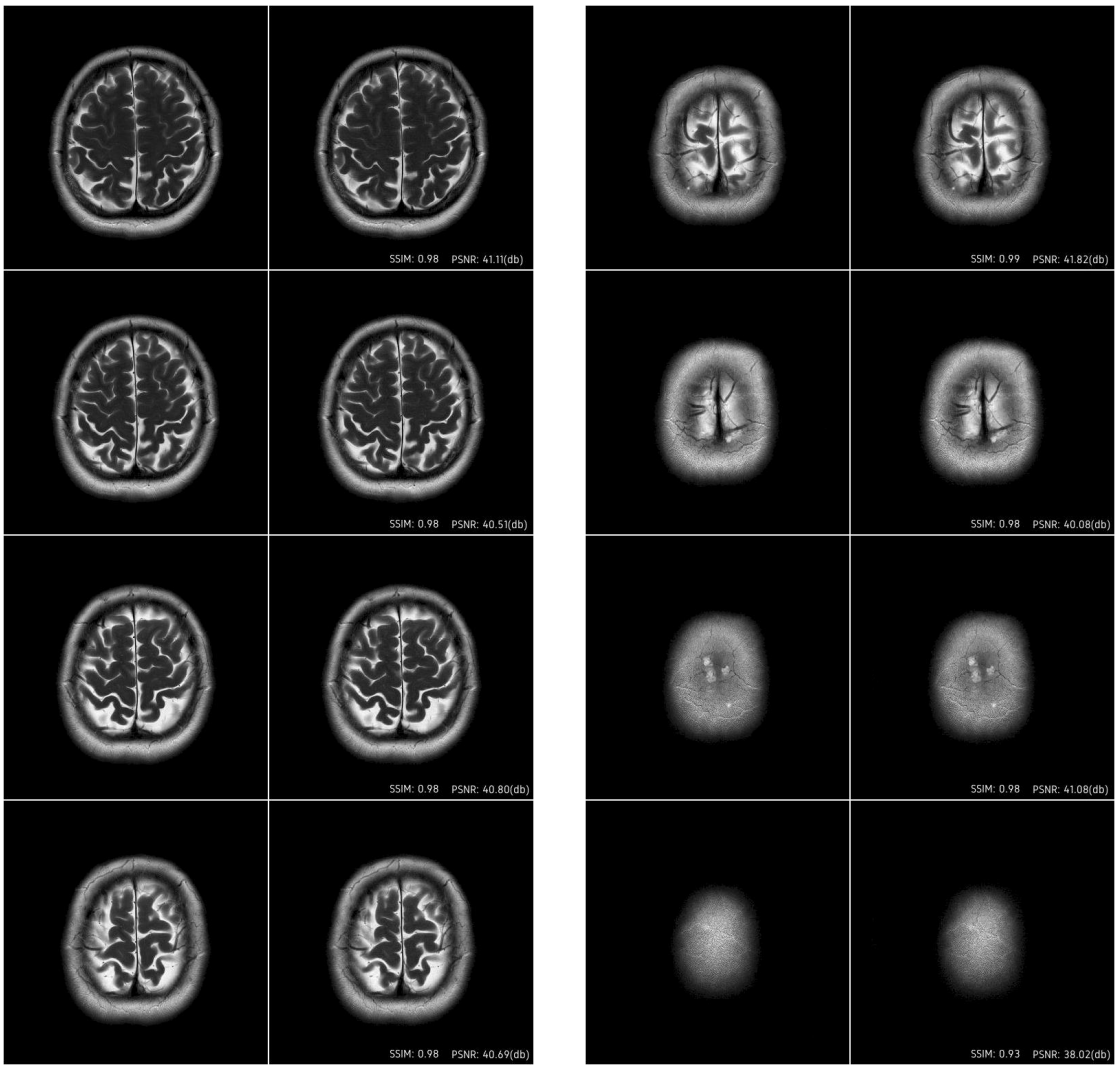
Appendix C

Langevin Dynamics Sample Sample Reconstructions



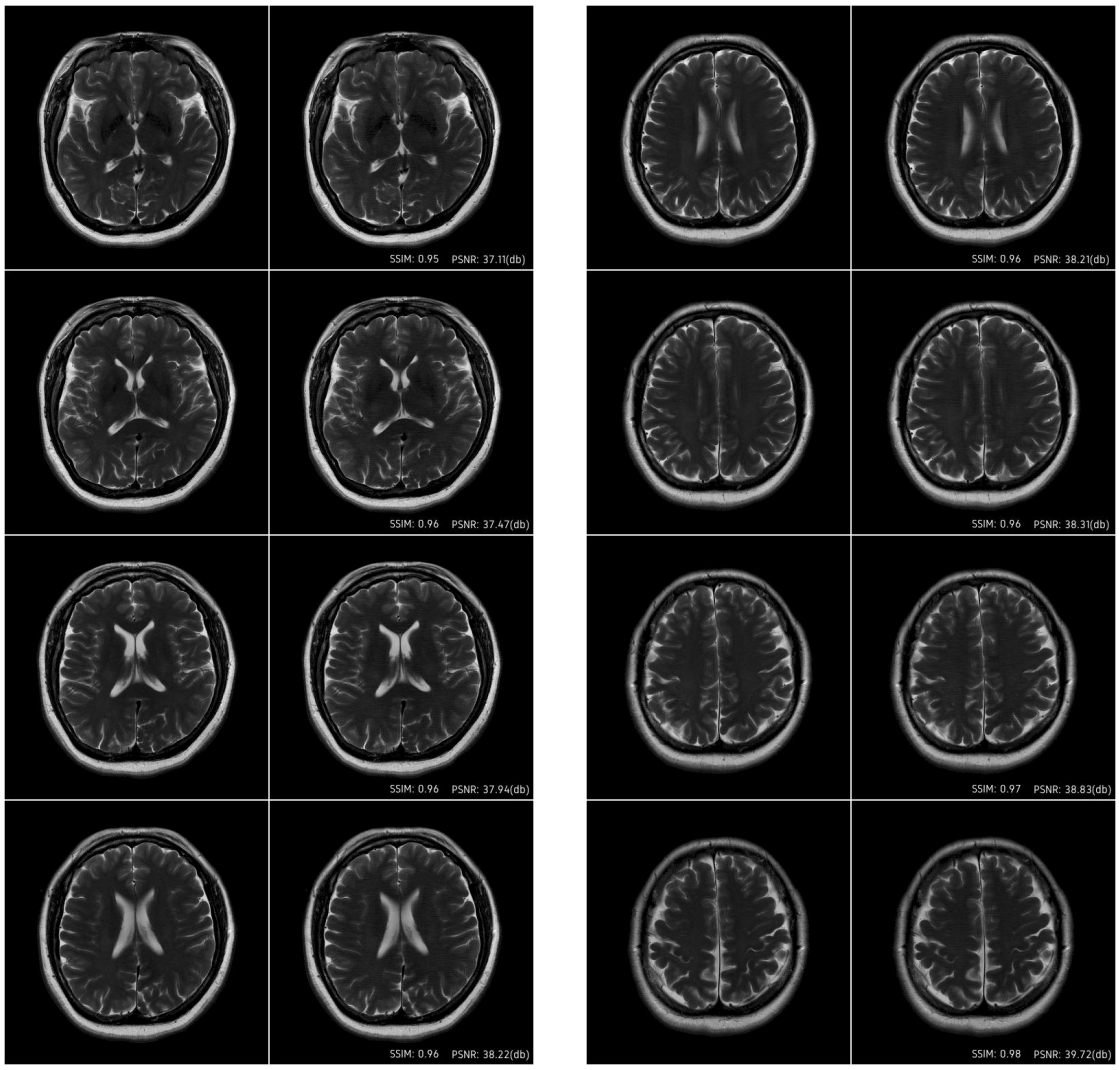
(a) (b)

Figure C.1: *R=2, 268 steps (part 1)*



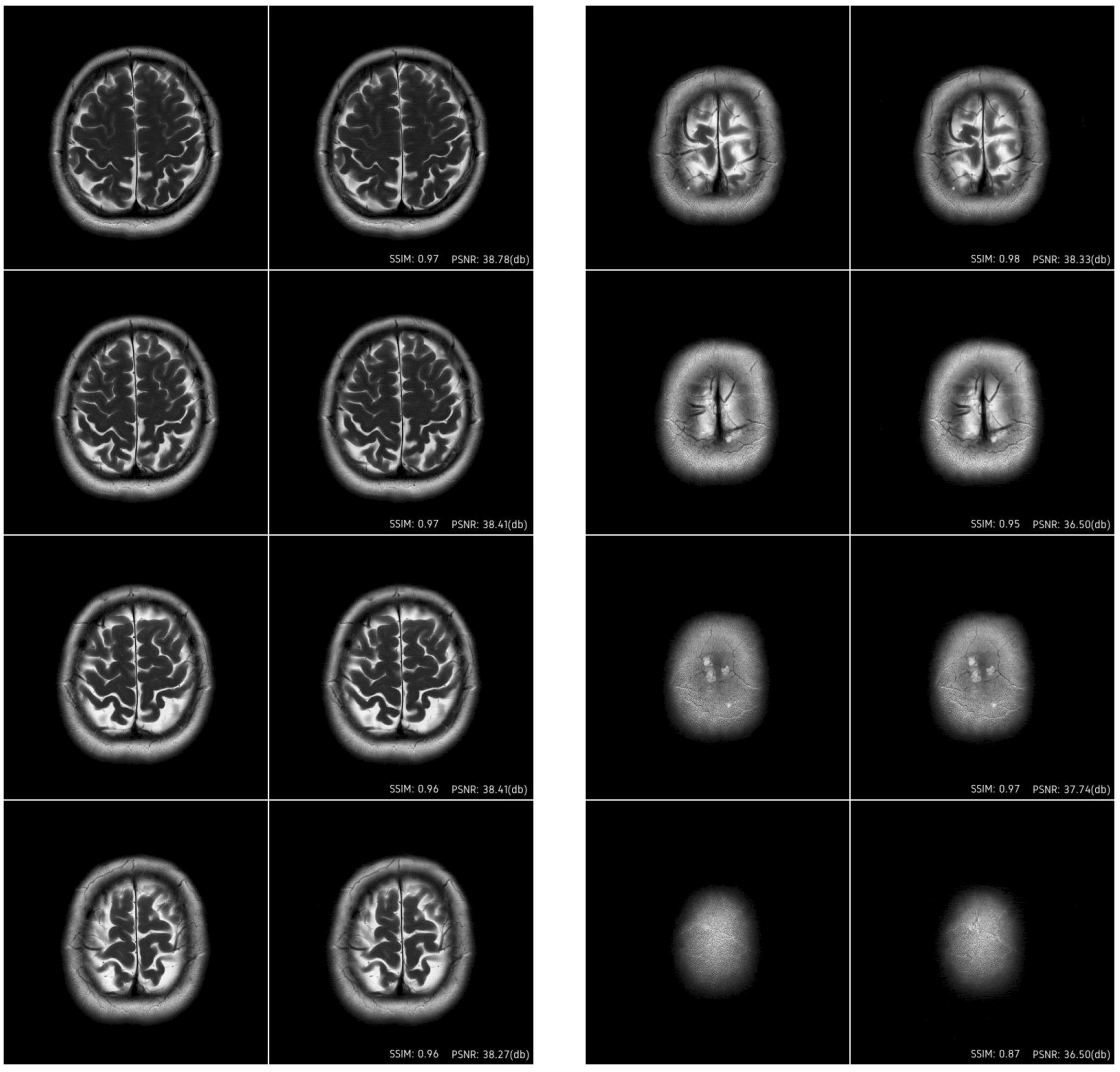
(a) (b)

Figure C.2: *R=2, 268 steps (part 1)*



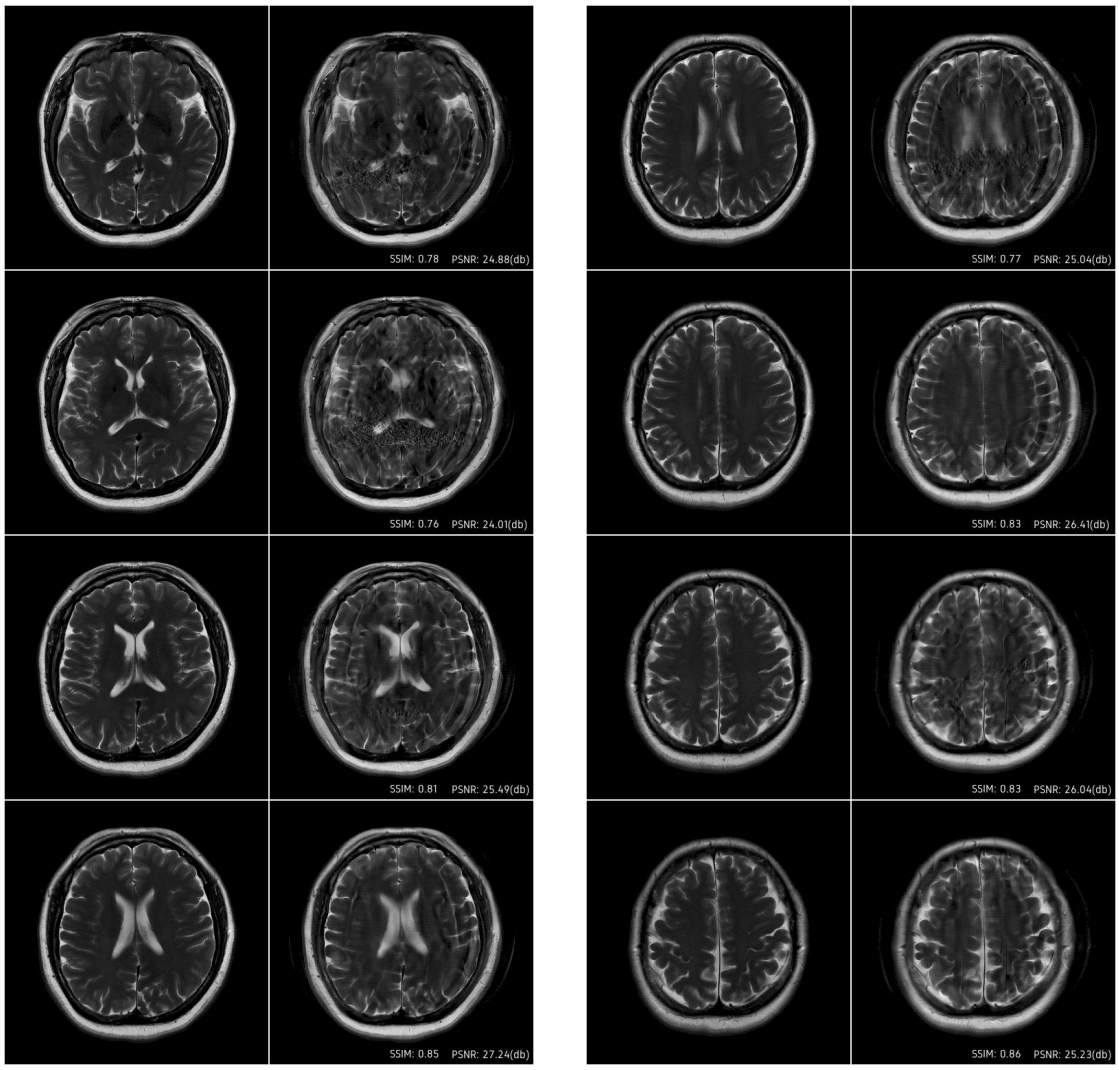
(a) (b)

Figure C.3: *R=4, 268 steps (part 1)*



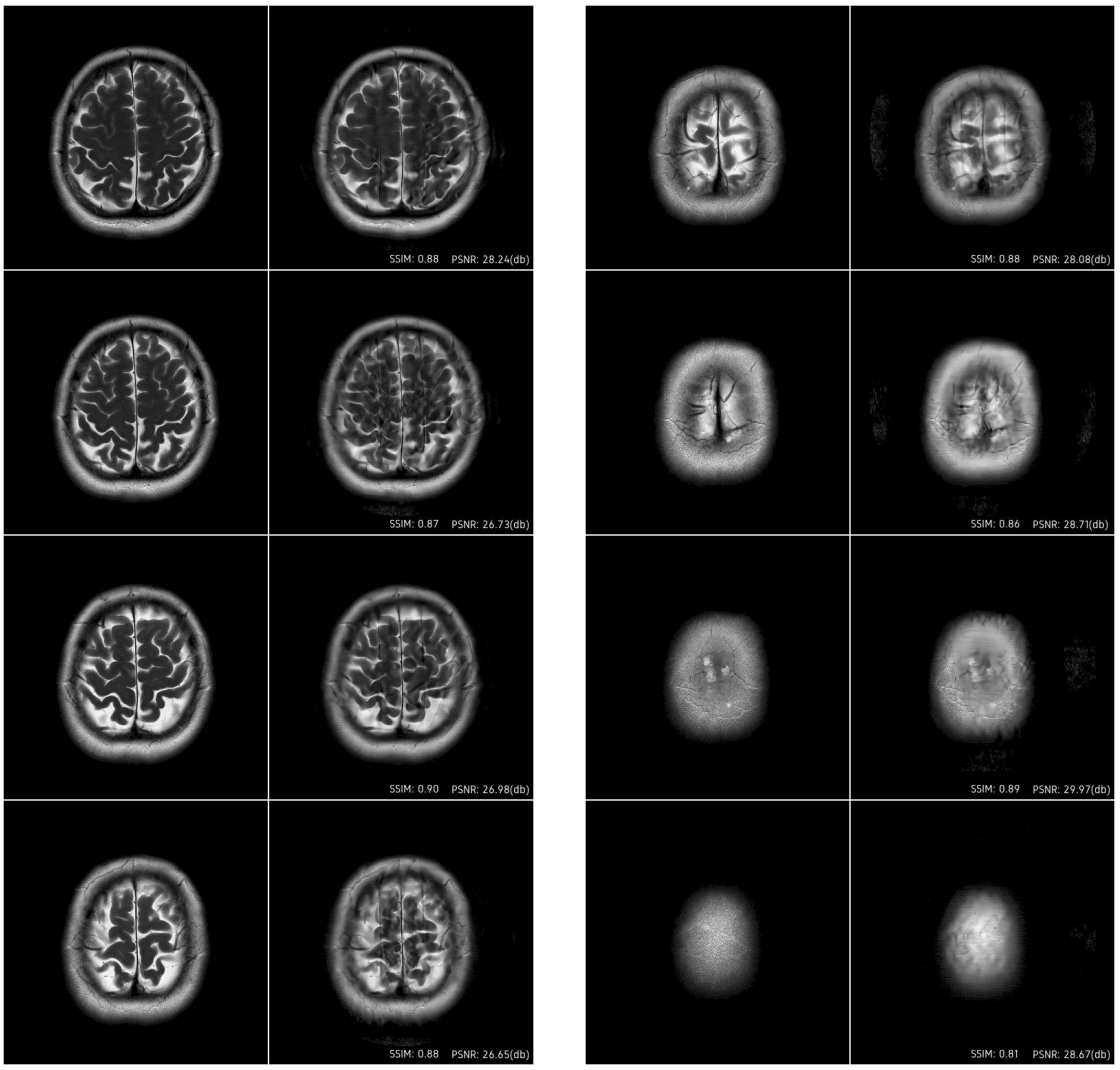
(a) (b)

Figure C.4: *R=4, 268 steps (part 1)*



(a) (b)

Figure C.5: *R=12, 268 steps (part 1)*



(a) (b)

Figure C.6: *R=12, 268 steps (part 1)*

Bibliography

[1] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan και Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. *CoRR*, abs/1503.03585, 2015.

[2] Calvin Luo. *Understanding Diffusion Models: A Unified Perspective*, 2022.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon και Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*. *International Conference on Learning Representations*, 2021.

[4] Jonathan Ho, Ajay Jain και Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. *Advances in Neural Information Processing Systems*H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan και H. Lin, επιμελητές, τόμος 33, σελίδες 6840–6851. Curran Associates, Inc., 2020.

[5] Jiaming Song, Chenlin Meng και Stefano Ermon. *Denoising Diffusion Implicit Models*. *CoRR*, abs/2010.02502, 2020.

[6] A. Hyvärinen. *Estimation of non-normalized statistical models by score matching*. *Journal of Machine Learning Research*, 2005.

[7] Zaccharie Ramzi, Benjamin Remy, Francois Lanusse, Jean Luc Starck και Philippe Ciuciu. *Denoising Score-Matching for Uncertainty Quantification in Inverse Problems*, 2020.

[8] Yang Song, Sahaj Garg, Jiaxin Shi και Stefano Ermon. *Sliced Score Matching: A Scalable Approach to Density and Score Estimation*. *CoRR*, abs/1905.07088, 2019.

[9] Yang Song. *Generative Modeling by Estimating Gradients of the Data Distribution*. *yang-song.net*, 2021.

[10] *Brownian Motion*. https://en.wikipedia.org/wiki/Brownian_motion. Access date: 26-08-2022.

[11] *Compressed Sensing*. https://en.wikipedia.org/wiki/Compressed_sensing. Access date: 26-08-2022.

[12] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis και Jonathan I Tamir. *Robust Compressed Sensing MRI with Deep Generative Priors*. *Advances in Neural Information Processing Systems*, 2021.

[13] Klaas Paul Pruessmann, Markus Weiger, Markus B. Scheidegger και Peter Boesiger. *SENSE: Sensitivity encoding for fast MRI*. *Magnetic Resonance in Medicine*, 42, 1999.

[14] Bahjat Kawar, Michael Elad, Stefano Ermon και Jiaming Song. *Denoising Diffusion Restoration Models*, 2022.

[15] Bradley Efron. *Tweedie's Formula and Selection Bias*. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. PMID: 22505788.

[16] H. Larochelle και I. Murray. *Compressed Sensing using Generative Models*. *International Conference on Artificial Intelligence and Statistics*, 2011.

[17] Gregor K. Murray I. Germain, M. και H. Larochelle. *Made: Masked autoencoder for distribution estimation*. *International Conference on Machine Learning*, 2015.

[18] Kalchbrenner N. Van Oord, A. και K. Kavukcuoglu. *Pixel recurrent neural networks*. *International Conference on Machine Learning*, 2016.

[19] Krueger D. Dinh, L. και Y. Bengio. *NICE: Non-linear independent components estimation*. *arXiv preprint arXiv:1410.8516*, 2014.

[20] Sohl Dickstein J. Dinh, L. και S. Bengio. *Density estimation using Real NVP*. *International Conference on Learning Representations*, 2017.

[21] Chopra S. Hadsell R. Ranzato M. LeCun, Y. και F. Huang. *A tutorial on energy-based learning*. *Predicting structured data, Vol 1(0)*, 2006.

[22] Y. Song και D.P. Kingma. *How to Train Your Energy-Based Models*. *arXiv preprint arXiv:2101.03288*, 2021.

[23] D.P. Kingma και M. Welling. *Auto-encoding variational bayes*. *International Conference on Learning Representations*, 2014.

[24] Mohamed S. Rezende, D.J. και D. Wierstra. *Stochastic backpropagation and approximate inference in deep generative models*. *International conference on machine learning*, 2014.

[25] S. Mohamed και B. Lakshminarayanan. *Learning in implicit generative models*. *arXiv preprint arXiv:1610.03483*, 2016.

[26] Pouget Abadie J. Mirza M. Xu B. Warde Farley D. Ozair S. Courville A. Goodfellow, I. και Y. Bengio. *Generative adversarial nets*. *Advances in neural information processing systems*, 2014.

[27] Goodfellow I. Zaremba W. Cheung V. Radford A. Salimans, T. και X. Chen. *Improved techniques for training gans*. *Advances in neural information processing systems*, 2016.

[28] Poole B. Pfau D. Metz, L. και J. Sohl-Dickstein. *Unrolled Generative Adversarial Networks*. *International Conference on Learning Representations*, 2017.

[29] Yang Song και Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. *Advances in Neural Information Processing Systems*, σελίδες 11895–11907, 2019.

[30] Yang Song και Stefano Ermon. *Improved Techniques for Training Score-Based Generative Models*. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan και Hsuan-Tien Lin, επιμελητές, 2020.

[31] *Structural Similarity Index Measure (SSIM)*. https://en.wikipedia.org/wiki/Structural_similarity. Access date: 26-08-2022.

[32] *Peak signal-to-noise ratio (PSNR)*. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio. Access date: 26-08-2022.

[33] Lee J. Liu, Q. και M. Jordan. *A kernelized Stein discrepancy for goodness-of-fit tests*. *International conference on machine learning*, 2016.

[34] Prafulla Dhariwal και Alexander Quinn Nichol. *Diffusion Models Beat GANs on Image Synthesis*. *Advances in Neural Information Processing Systems*A. Beygelzimer, Y. Dauphin, P. Liang και J. Wortman Vaughan, επιμελητές, 2021.

[35] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi και William Chan. *WaveGrad: Estimating Gradients for Waveform Generation*. *International Conference on Learning Representations*, 2021.

[36] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao και Bryan Catanzaro. *DiffWave: A Versatile Diffusion Model for Audio Synthesis*. *International Conference on Learning Representations*, 2021.

[37] Vovk I. Gogoryan V. Sadekova T. Popov, V. και M. Kudinov. *Grad-tts: A diffusion probabilistic model for text-to-speech*. 2021.

[38] Yang G. Averbuch Elor H. Hao Z. Belongie S. Snavely N. Cai, R. και B. Hariharan. *Learning Gradient Fields for Shape Generation*. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[39] Engel J. Hawthorne C. Mittal, G. και I. Simon. *Symbolic Music Generation with Diffusion Models*. *arXiv preprint arXiv:2103.16091*, 2021.

[40] Max Welling και Yee Whye Teh. *Bayesian Learning via Stochastic Gradient Langevin Dynamics*. *ICML*Lise Getoor και Tobias Scheffer, επιμελητές, σελίδες 681–688. Omnipress, 2011.

[41] *Diffusion*. https://en.wikipedia.org/wiki/Diffusion. Access date: 26-08-2022.

[42] William Feller. *On the Theory of Stochastic Processes, with Particular Reference to Applications*. *Proceedings of the [first] Berkeley Symposium on Mathematical Statistics and Probability*, 1949.

[43] Alex Nichol και Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. *CoRR*, abs/2102.09672, 2021.

[44] Brian D.O. Anderson. *Reverse-time diffusion equation models*. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[45] *Lasso (statistics)*. https://en.wikipedia.org/wiki/EulerâĂŞMaruyama_method. Access date: 26-08-2022.

[46] *l1 MAgic MATLAB routines*. https://github.com/scgt/l1magic. Access date: 26-08-2022.

[47] Michael Lustig, David Donoho και John M. Pauly. *Sparse MRI: The application of compressed sensing for rapid MR imaging*. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[48] Mark Murphy και Miki Lustig. *Chapter 45 - 1 Minimization in 1-SPIRiT Compressed Sensing MRI Reconstruction*. *GPU Computing Gems Emerald Edition*Wenmei W. Hwu, επιμελητής, Applications of GPU Computing Series, σελίδες 723–735. Morgan Kaufmann, Boston, 2011.

[49] Feng Huang, Sathya Vijayakumar, Yu Li, Sarah Hertel και George R. Duensing. *A software channel compression technique for faster reconstruction with many channels*. *Magnetic Resonance Imaging*, 26(1):133–141, 2008.

[50] Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Rémi Tachet des Combes και Ioannis Mitliagkas. *Adversarial score matching and improved sampling for image generation*, 2020.

[51] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson και Yvonne W. Lui. *fastMRI: An Open Dataset and Benchmarks for Accelerated MRI*. 2018.

[52] Guosheng Lin, Anton Milan, Chunhua Shen και Ian D. Reid. *RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation*. *CoRR*, abs/1611.06612, 2016.

[53] Daniel Watson, Jonathan Ho, Mohammad Norouzi και William Chan. *Learning to Efficiently Sample from Diffusion Probabilistic Models*. *CoRR*, abs/2106.03802, 2021.

[54] Zhifeng Kong και Wei Ping. *On Fast Sampling of Diffusion Probabilistic Models*. *CoRR*, abs/2106.00132, 2021.

[55] Robin San-Roman, Eliya Nachmani και Lior Wolf. *Noise Estimation for Generative Diffusion Models*. *CoRR*, abs/2104.02600, 2021.

[56] Tim Salimans και Jonathan Ho. *Progressive Distillation for Fast Sampling of Diffusion Models*. *CoRR*, abs/2202.00512, 2022.

[57] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet και Mohammad Norouzi. *Image Super-Resolution via Iterative Refinement*, 2021.

[58] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi και Tim Salimans. *Cascaded Diffusion Models for High Fidelity Image Generation*. *CoRR*, abs/2106.15282, 2021.