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

Mitigating Exposure Bias in Discriminator Guided Diffusion Models

Διπλωματικη Εργασια

του

ΕΛΕΥΘΕΡΙΟΥ Γ. ΤΣΩΝΗ

Επιβλέπων: Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Μέλος Ε.ΔΙ.Π. Ε.Μ.Π.

Αθήνα, Νοέμβριος 2023

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

Mitigating Exposure Bias in Discriminator Guided Diffusion Models

Διπλωματικη Εργασια

του

ΕΛΕΥΘΕΡΙΟΥ Γ. ΤΣΩΝΗ

Επιβλέπων: Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Μέλος Ε.ΔΙ.Π. Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Νοεμβρίου 2023.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Αθανάσιος Βουλόδημος	Γεώργιος Στάμου	Στέφανος Κόλλιας
Επίκουρος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2023

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.

(Υπογραφή)

.....
Ελευθέριος Τσώνης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

1η Νοεμβρίου 2023

Περίληψη

Τα μοντέλα διάχυσης έχουν επιδείξει αξιοσημείωτες επιδόσεις στη δημιουργία εικόνων. Ωστόσο, οι υψηλές υπολογιστικές απαιτήσεις τους για εκπαίδευση έχουν οδηγήσει σε συνεχείς προσπάθειες βελτίωσης της ποιότητας των παραγόμενων εικόνων μέσω τροποποιήσεων στη διαδικασία δειγματοληψίας. Μια πρόσφατη προσέγγιση, γνωστή ως Καθοδήγηση Διαχωριστή, επιδιώκει να γεφυρώσει το χάσμα μεταξύ του score του μοντέλου και του score των δεδομένων, ενσωματώνοντας έναν βοηθητικό όρο, που προέρχεται από ένα δίκτυο διαχωριστή. Δείχνουμε ότι παρά τη σημαντική βελτίωση στην ποιότητα των δειγμάτων, η τεχνική αυτή δεν έχει επιλύσει το ζήτημα της μεροληψίας έκθεσης. Η μεροληψία έκθεσης αναφέρεται στην διαφορά μεταξύ των δεδομένων εισόδου κατά τη φάση εκπαίδευσης και τη φάση δειγματοληψίας και οδηγεί σε μειωμένη ποιότητα δειγμάτων στα μοντέλα διάχυσης. Προτείνουμε το SEDM-G++, το οποίο ενσωματώνει μια τροποποιημένη προσέγγιση δειγματοληψίας, συνδυάζοντας την Καθοδήγηση Διαχωριστή και την Κλιμάκωση Έψιλον. Η προτεινόμενη προσέγγισή μας ξεπερνάει το τρέχον state-of-the-art στη δημιουργία συνθετικών εικόνων.

Λέξεις Κλειδιά

Μοντέλα Διάχυσης, Score-Based Παραγωγικά μοντέλα, Στοχαστικές Διαφορικές Εξισώσεις, Παραγωγή εικόνων, Όραση Υπολογιστών

Abstract

Diffusion Models have demonstrated remarkable performance in image generation. However, their demanding computational requirements for training have prompted ongoing efforts to enhance the quality of generated images through modifications in the sampling process. A recent approach, known as Discriminator Guidance, seeks to bridge the gap between the model score and the data score by incorporating an auxiliary term, derived from a discriminator network. We show that despite significantly improving sample quality, this technique has not resolved the persistent issue of Exposure Bias. Exposure bias refers to the discrepancy between the input data during training and inference phases and leads to diminished sample quality in diffusion models. We propose SEDM-G++, which incorporates a modified sampling approach, combining Discriminator Guidance and Epsilon Scaling. Our proposed framework outperforms the current state-of-the-art in unconditional image generation.

Keywords

Diffusion Models, Score-Based Generative Models, Stochastic Differential Equations, Generative AI, Image Generation, Computer Vision

στην οικογένειά μου

Ευχαριστίες

Η ολοκλήρωση αυτής της διπλωματικής εργασίας σηματοδοτεί το πέρας της πενταετούς φοίτησής μου στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π. Θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλαν σε αυτήν την πορεία.

Αρχικά, τους επιβλέποντες της εργασίας μου, τον κ. Θάνο Βουλόδημο και την κα. Παρασκευή Τζούβελη. Σας ευχαριστώ για την εμπιστοσύνη που μου δείξατε και την ελευθερία που μου δώσατε να επιλέξω την κατεύθυνση της έρευνας. Μαζί με τους επιβλέποντες, θα ήθελα να ευχαριστήσω και τον διευθυντή του Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης, κ. Γιώργο Στάμου. Είναι τιμή μου να συνεργάζομαι με αυτούς τους τρεις εξαιρετικούς ανθρώπους, που αποτελούν πρότυπα για εμένα, τόσο ακαδημαϊκά, όσο και λόγω του ήθους και του χαρακτήρα τους. Είμαι ευγνώμων για το εκπαιδευτικό αλλά και για το ανθρώπινο ενδιαφέρον σας.

Θα ήθελα επίσης να ευχαριστήσω την οικογένεια μου. Τους γονείς μου, Λένα και Γιώργο, που στηρίζουν κάθε βήμα μου και χωρίς αυτούς δεν θα μπορούσα να γίνω ο άνθρωπος που είμαι σήμερα. Την αγαπημένη μου αδερφή, Ιωάννα, που πάντα βρίσκεται στο πλευρό μου και με στηρίζει.

Τέλος, ένα μεγάλο ευχαριστώ στους συνοδοιπόρους των φοιτητικών ετών, τους κολλητούς μου φίλους, Αλέξανδρο, Βαγγέλη, Βασίλη, Δήμητρα και Θοδωρή. Οι αμέτρητες ώρες που περάσαμε μαζί, διαβάζοντας, γελώντας, διασκεδάζοντας, μας εφοδίασαν με αναμνήσεις για μια ζωή.

Σας ευχαριστώ!

Αθήνα, Νοέμβριος 2023
Ελευθέριος Τσώνης

Table of Contents

List of Figures

List of Tables

Part I

Εκτεταμένη Περίληψη στα Ελληνικά

Chapter 1

Εισαγωγή στα Μοντέλα Διάχυσης

Τα μοντέλα διάχυσης (diffusion models), τα οποία προτάθηκαν αρχικά από τους Sohl-Dickstein *et al.* το 2015 [6], έχουν διακριθεί σε διάφορους τομείς, συμπεριλαμβανομένης της παραγωγής συνθετικών εικόνων, συνθετικού ήχου [7, 8] και συνθετικών βίντεο [9, 10, 11]. Στον τομέα της σύνθεσης εικόνων, τα μοντέλα διάχυσης γνώρισαν σημαντική πρόοδο τα τελευταία χρόνια μέσω της συνεισφοράς των Song και Ermon (2019) [4], Ho *et al.* (2020) [12], και Nichol και Dhariwal (2021) [13]. Το 2021, οι Song *et al.* παρουσίασαν μια νέα προσέγγιση που ενοποιεί τα score-based μοντέλα και τα Denoising Diffusion Probabilistic Models (DDPMs) χρησιμοποιώντας στοχαστικές διαφορικές εξισώσεις (SDEs) [3]. Επιπλέον, οι Karras *et al.* (2022) παρουσίασαν μια ολοκληρωμένη διερεύνηση του χώρου σχεδιασμού των μοντέλων διάχυσης και εισήγαγαν το μοντέλο EDM [14], το οποίο εφάρμοσε μια σειρά από βελτιστοποιήσεις τόσο στη διαδικασία δειγματοληψίας όσο και στη διαδικασία εκπαίδευσης, οδηγώντας σε σημαντική βελτίωση της απόδοσης και της ποιότητας των δειγμάτων.

1.1 Μοντέλα Διάχυσης

Τα μοντέλα διάχυσης παρουσιάστηκαν για πρώτη φορά το 2015 [6]. Εμπνευσμένα από τη θερμοδυναμική μη ισορροπίας, τα μοντέλα διάχυσης εισάγουν σταδιακά θόρυβο στα δεδομένα, σε μια επαναληπτική διαδικασία ευθείας διάχυσης (forward diffusion process). Ο στόχος είναι να μάθουμε μια διαδικασία αντίστροφης διάχυσης (reverse diffusion process), η οποία αναιρεί την προσθήκη θορύβου. Έτσι, το μοντέλο καθίσταται ικανό να παράγει ρεαλιστικά δείγματα από θόρυβο.

1.1.1 Διαδικασία ευθείας διάχυσης

Ας θεωρήσουμε την κατανομή δεδομένων $x_0 \sim q(x_0)$. Στη διαδικασία της ευθείας διάχυσης, θόρυβος προστίθεται σταδιακά σε ένα καθαρό δεδομένο, μέχρι να καταστραφεί η δομή του και το δεδομένο να μετατραπεί σε καθαρό θόρυβο. Με άλλα λόγια, η ευθεία διάχυση χαρακτηρίζεται από μια αλυσίδα Markov, η οποία παράγει μια ακολουθία τυχαίων μεταβλητών $x_1, x_2, \ldots x_T$ με πυρήνα μετάβασης $q(x_t|x_{t-1})$. Ορίζουμε ένα πρόγραμμα διακύμανσης $\{\beta_t \in (0,1)\}_{t=0}^{T}$, έτσι ώστε ο θόρυβος που διαταράσσει τα δεδομένα σε κάθε διακριτό χρονικό βήμα t να είναι μια ισότροπη Γκαουσιανή με διακύμανση β_t. Η αλυσίδα Markov

ορίζεται από τον πυρήνα μετάβασης:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1.1}$$

Ο πυρήνας μετάβασης είναι συχνά χειροποίητος για να μετασχηματίσει την αρχική κατανομή δεδομένων σε μια προηγούμενη κατανομή. Έτσι, η διαδικασία της ευθείας διάχυσης μπορεί να υλοποιηθεί εύκολα. Το μοντέλο διάχυσης στοχεύει στην εκμάθηση της κατανομής της αντίστροφης διαδικασίας $q(x_{t-1}|x_t)$, η οποία είναι άγνωστη.

Έστω x_t που δειγματοληπτείται από το $q(x_t|x_{t-1})$ στο χρονικό βήμα t. Αποδεικνύεται από το Λήμμα 6.1 και την Πρόταση 6.1, ότι το x_t μπορεί να εκφραστεί άμεσα ως γραμμικός συνδυασμός του x_0 και μιας μεταβλητής θορύβου ϵ:

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

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

Οι κατανομές $\mathcal{N}(x_t; \sqrt{a_t}x_{t-1}, (1-a_t)\mathbf{I})$ και $\mathcal{N}(x_t; \sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)\mathbf{I})$, όπου $a_t = 1-\beta_t$, είναι ισοδύναμες. Αυτό είναι σημαντικό, καθώς επιτρέπει τη δειγματοληψία από οποιαδήποτε ενδιάμεση κατανομή της διαδικασίας διάχυσης προς τα εμπρός σε ένα μόνο βήμα.

1.1.2 Διαδικασία αντίστροφης διάχυσης

Η διαδικασία αντίστροφης διάχυσης μπορεί να χρησιμοποιηθεί για τη δημιουργία νέων δειγμάτων δεδομένων. Τα μοντέλα διάχυσης ξεκινούν με τη δημιουργία ενός αρχικού γκαουσιανού δείγματος $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Στη συνέχεια, ο θόρυβος αφαιρείται σταδιακά με την διάσχιση μιας μαθήσιμης (learnable) αλυσίδας Markov στην αντίστροφη χρονική κατεύθυνση. Οι Sohl-Dickstein *et al.* δείχνουν ότι αν το β_t είναι αρκετά μικρό, τότε το $q(x_{t-1}|x_t)$ θα είναι επίσης μια γκαουσιανή κατανομή [6]. Για να εκτελέσουμε αντίστροφη διάχυση, θα πρέπει να μοντελοποιήσουμε τους πυρήνες των αντίστροφων χρονικών βημάτων. Η εκτίμηση του $q(x_{t-1}|x_t)$ είναι ανέφικτη. Αντ' αυτού, χρησιμοποιείται ένας learnable πυρήνας μετάβασης p_∂ για την προσέγγιση των πιθανοτήτων. Ο πυρήνας έχει την ακόλουθη μορφή:

$$p_\partial(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\partial(x_t, t), \Sigma_\partial(x_t, t)) \tag{1.2}$$

όπου το ∂ συμβολίζει τις παραμέτρους του μοντέλου και η μέση τιμή $\mu_\partial(x_t, t)$ και η διακύμανση $\Sigma_\partial(x_t, t)$ παραμετροποιούνται από νευρωνικά δίκτυα.

Το κλειδί για να λειτουργήσει η διαδικασία δειγματοληψίας είναι η εκπαίδευση της αντίστροφης αλυσίδας Markov έτσι ώστε να μιμείται τη χρονική αντιστροφή της ευθείας αλυσίδας Markov. Με άλλα λόγια, η παράμετρος ∂ πρέπει να ρυθμιστεί ώστε να διασφαλιστεί ότι η συμπεριφορά της αντίστροφης αλυσίδας Markov μοιάζει με εκείνη της ευθείας διαδικασίας. Η προσαρμογή αυτή πρέπει να είναι τέτοια ώστε η κοινή κατανομή της αντίστροφης αλυσίδας Markov $p_\partial(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\partial(x_{t-1}|x_t)$ (Εξ. 6.4) να προσεγγίζει στενά εκείνη της ευθείας διαδικασίας $q(x_{0:T}) = q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1})$.

Το $q(x_{t-1}|x_t)$ είναι υπολογίσιμο, αν το εξαρτήσουμε στο x_0. Λαμβάνουμε:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t\mathbf{I}) \tag{1.3}$$

Μπορούμε να αναδιατάξουμε την εξίσωση στην Πρόταση 6.1, για να δείξουμε ότι το x_0 μπορεί να εκφραστεί με την ακόλουθη μορφή:

$$x_0 = \frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1-\bar{a}_t}\epsilon_t) \tag{1.4}$$

Εφαρμόζοντας τον κανόνα του Bayes και την Εξ. 1.4, προκύπτουν η μέση τιμή και η διακύμανση:

$$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right) \tag{1.5}$$

$$\tilde{\beta}_t = \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\beta_t \tag{1.6}$$

1.1.3 Στόχος εκπαίδευσης

Για να εκπαιδεύσουμε την αντίστροφη αλυσίδα Markov ώστε να ταιριάζει με τη χρονική αντιστροφή της ευθείας αλυσίδας Markov, πρέπει να προσαρμόσουμε την παράμετρο θ, έτσι ώστε η κοινή αντίστροφη κατανομή:

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{1.7}$$

να προσεγγίζει στενά την κοινή κατανομή της ευθείας διαδικασίας:

$$q(x_{0:T}) = q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1.8}$$

Χρησιμοποιούμε το μοντέλο μας για να προσεγγίσουμε την τιμή $\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right)$. Δεδομένου ότι έχουμε πρόσβαση στο x_t κατά τη διάρκεια της εκπαίδευσης, εκπαιδεύουμε το μοντέλο μας για να προβλέψουμε τον όρο του γκαουσιανού θορύβου από την είσοδο x_t, στο χρονικό βήμα t:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_\theta(x_t, t)\right)$$

Το x_{t-1} γίνεται:

$$x_{t-1} = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_\theta(x_t, t)\right), \Sigma_\theta(x_t, t))$$

Οι Ho *et al.* [12] παραμετροποιούν τον όρο απώλειας (loss term) L_t ώστε να ελαχιστοποιείται η διαφορά από το $\tilde{\mu}_t$:

$$L_t = \mathbb{E}_{x_0,\epsilon}\left[\frac{(1-a_t)^2}{2a_t(1-\bar{a}_t)\|\Sigma_\theta(x_t,t)\|_2^2}\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}\epsilon_t, t)\|^2\right]$$

Διαπιστώνεται ότι η παράλειψη του όρου στάθμισης από τον στόχο εκπαίδευσης είναι ευκολότερη στην εφαρμογή και ωφελεί την ποιότητα των δειγμάτων:

$$L_t^{simple} = \mathbb{E}_{t,x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}\epsilon_t, t)\|^2\right]$$

όπου το t είναι ομοιόμορφο στο $[1, T]$.

1.1.4 Denoising Diffusion Implicit Models

Έχουμε διαπιστώσει ότι τα μοντέλα διάχυσης παράγουν δείγματα χρησιμοποιώντας μια αλυσίδα Markov που ξεκινά από ένα δείγμα καθαρού θορύβου και το αποθορυβοποιεί προοδευτικά, δημιουργώντας ένα νέο δείγμα. Αυτή η αλυσίδα Markov λαμβάνεται με την κατά προσέγγιση αντιστροφή της διαδικασίας ευθείας διάχυσης, η οποία αποτελείται από έως και μερικές χιλιάδες βήματα, γεγονός που απαιτεί πολλές επαναλήψεις. Λόγω αυτού του περιορισμού, τα μοντέλα διάχυσης ήταν πολύ πιο αργά, σε σύγκριση με άλλα σύγχρονα παραγωγικά μοντέλα, όπως τα Generative Adversarial Networks (GANs).

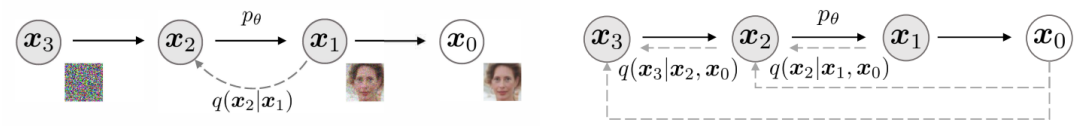

Figure 1.1. *Σύγκριση μοντέλων διάχυσης (αριστερά) και μη μαρκοβιανών μοντέλων (δεξιά) [1].*

Οι Song *et al.* εισήγαγαν τα Denoising Diffusion Implicit Models (DDIMs) [1], τα οποία ήταν τα πρώτα implicit πιθανοτικά μοντέλα που μπορούσαν να παράγουν δείγματα υψηλής ποιότητας. Βελτίωσαν την ταχύτητα δειγματοληψίας επεκτείνοντας το αρχικό DDPM σε μη μαρκοβιανές περιπτώσεις.

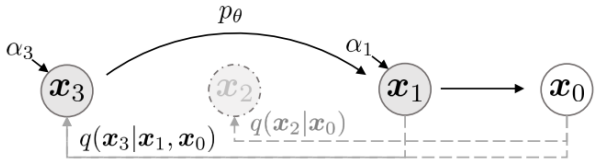

Figure 1.2. *Επιταχυμενη παραγωγή δειγμάτων με χρήση DDIM [1].*

1.2 Score-Based Παραγωγικά Μοντέλα

Ένα άλλο σημαντικό μέλος της οικογένειας των μοντέλων διάχυσης είναι τα Score-Based παραγωγικά μοντέλα (SGM) [4]. Δεδομένης της κατανομής δεδομένων $p_{data}(x)$, ορίζουμε τη συνάρτηση score ως την κλίση του λογαρίθμου της συνάρτησης πυκνότητας πιθανότητας, γνωστή και ως συνάρτηση score (Stein). Η βασική αρχή των SGMs είναι η εκτίμηση της συνάρτησης score, χρησιμοποιώντας ένα βαθύ νευρωνικό δίκτυο, και η δημιουργία δειγμάτων χρησιμοποιώντας προσεγγίσεις δειγματοληψίας βασισμένες στο σκορ.

Τα score-based μοντέλα έχουν επιτύχει κορυφαία αποτελέσματα σε tasks όπως η παραγωγή συνθετικών εικόνων [4, 15, 14, 12], και ήχου [8, 7].

1.2.1 Συνάρτηση score και score matching για την εκτίμηση της

Ας υποθέσουμε ότι το σύνολο δεδομένων μας αποτελείται από i.i.d. δείγματα $\{x_i \in \mathbb{R}^D\}_{i=1}^N$ από μια κατανομή δεδομένων $p_{data}(x)$, η οποία είναι άγνωστη. Η συνάρτηση score μιας

πυκνότητας πιθανότητας $p(x)$ ορίζεται ως $\nabla_x \log p(x)$. Είναι ουσιαστικά ένα διανυσματικό πεδίο, που δείχνει προς την κατεύθυνση όπου ο λογάριθμος της συνάρτησης πυκνότητας πιθανότητας αυξάνεται τάχιστα. Το δίκτυο score $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$ είναι ένα νευρωνικό δίκτυο με παράμετρο την θ, το οποίο θα εκπαιδευτεί για να προσεγγίσει το score του $p_{data}(x)$.

Το score matching χρησιμοποιήθηκε αρχικά για την εκμάθηση μη κανονικοποιημένων στατιστικών μοντέλων με βάση i.i.d. δείγματα από μια κατανομή δεδομένων. Μπορεί να επαναχρησιμοποιηθεί στην εκπαίδευση ενός δικτύου score $s_\theta(x)$ για την εκτίμηση του $\nabla_x \log p_{data}(x)$ χωρίς να εκτιμηθεί πρώτα το $p_{data}(x)$. Ο στόχος είναι η ελαχιστοποίηση

$$\frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\|s_\theta(x) - \nabla_x \log p_{data}(x)\|_2^2\right] \tag{1.9}$$

η οποία είναι ισοδύναμη με την ακόλουθη:

$$\mathbb{E}_{p_{data}(x)}\left[\mathrm{tr}(\nabla_x s_\theta(x)) + \frac{1}{2}\|s_\theta(x)\|_2^2\right] \tag{1.10}$$

όπου $\nabla_x s_\theta(x)$ συμβολίζει τη Ιακωβιανή του $s_\theta(x)$.

Ενώ η αναμενόμενη τιμή για το $p_{data}(x)$ μπορεί να εκτιμηθεί γρήγορα χρησιμοποιώντας δείγματα δεδομένων, το score matching δεν είναι επεκτάσιμο σε βαθιά δίκτυα και δεδομένα υψηλής διαστατικότητας λόγω του υπολογιστικού κόστους του $\mathrm{tr}(\nabla_x s_\theta(x))$. Για να παρακάμψουμε αυτό το πρόβλημα, παρουσιάζουμε δύο δημοφιλείς μεθόδους για score matching μεγάλης κλίμακας.

Denoising score matching Το denoising score matching [16] θορυβοποιεί το σημείο δεδομένων x με μια προκαθορισμένη κατανομή θορύβου $q_\sigma(\tilde{x}|x)$ και χρησιμοποιεί score matching για να εκτιμήσει το σκορ της θορυβοποιημένης κατανομής δεδομένων, $q_\sigma(\tilde{x}) \triangleq \int q_\sigma(\tilde{x}|x)p_{data}(x)dx$. Ο στόχος αποδεικνύεται ισοδύναμος με:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{data}(x)}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2\right] \tag{1.11}$$

Το βέλτιστο δίκτυο score, $s_{\theta^*}(x)$, ικανοποιεί $s_{\theta^*}(x) = \nabla_x \log q_\sigma(x)$ σχεδόν σίγουρα, ενώ $s_{\theta^*}(x) = \nabla_x \log q_\sigma(x) \approx \nabla_x \log p_{data}(x)$ ισχύει μόνο όταν ο θόρυβος είναι αρκετά μικρός, έτσι ώστε $q_\sigma(x) \approx p_{data}(x)$.

Sliced score matching το sliced score matching [17] χρησιμοποιεί τυχαίες προβολές για να προσεγγίσει το $\mathrm{tr}(\nabla_x s_\theta(x))$ στο score matching. Ο στόχος γίνεται:

$$\mathbb{E}_{p_v}\mathbb{E}_{p_{data}}\left[v^T \nabla_x s_\theta(x) v + \frac{1}{2}\|s_\theta(x)\|_2^2\right] \tag{1.12}$$

Ο όρος $v^T \nabla_x s_\theta(x) v$ μπορεί να υπολογιστεί αποδοτικά με forward mode auto-differentiation. Σε αντίθεση με το denoising score matching, το οποίο εκτιμά τη συνάρτηση score των θορυβοποιημένων δεδομένων, το sliced score matching εκτιμά τη συνάρτηση βαθμολογίας της αρχικής κατανομής. Το sliced score matching απαιτεί περίπου τέσσερις φορές περισσότερους υπολογισμούς από το denoising score matching.

1.2.2 Δειγματοληψία με Langevin Dynamics

Τα Langevin Dynamics είναι ικανά να παράξουν δείγματα από μια πυκνότητα πιθανότητας $p(x)$, χρησιμοποιώντας μόνο τη συνάρτηση score $\nabla_x \log p(x)$. Αυτός είναι ο λόγος για τον οποίο η εκτίμηση της συνάρτησης score είναι τόσο σημαντική στα score-based παραγωγικά μοντέλα. Με ένα αρχικό σημείο δεδομένων $\tilde{x}_0 \sim \pi(x)$ (π είναι μια εκ των προτέρων κατανομή) και μέγεθος βήματος $\epsilon > 0$, τα Langevin Dynamics υπολογίζουν αναδρομικά την ακόλουθη ποσότητα:

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x \log p(\tilde{x}_{t-1}) + \sqrt{\epsilon} z_t \tag{1.13}$$

όπου $z_t \sim N(\mathbf{0}, \mathbf{I})$. Όταν $\epsilon \to 0$ και $T \to \infty$, η κατανομή του $\tilde{x}_T$ ισούται με $p(x)$. Υπό ορισμένες συνθήκες κανονικότητας [18], η $\tilde{x}_T$ γίνεται ακριβές δείγμα από την $p(x)$. Στην πράξη, θέτουμε το ϵ σε μικρό αριθμό και το T σε μεγάλο αριθμό.

Η διαδικασία της δειγματοληψίας από την Εξ. 1.13 βασίζεται στη συνάρτηση score $\nabla_x \log p(x)$. Επομένως, για να λάβουμε δείγματα από το $p_{data}(x)$, μπορούμε αρχικά να εκπαιδεύσουμε το δίκτυο score μας για να προσεγγίσουμε το $s_\theta(x) \approx \nabla_x \log p_{data}(x)$. Στη συνέχεια, μπορούμε να χρησιμοποιήσουμε αυτή την προσέγγιση σε συνδυασμό με τα Langevin dynamics για την παραγωγή δειγμάτων. Αυτή η θεμελιώδης ιδέα αποτελεί τη βασική αρχή των score-based παραγωγικών μοντέλων.

1.2.3 Noise Conditional Score Networks (NCSNs)

Οι Song και Ermon [4] παρατήρησαν ότι η θορυβοποίηση των δεδομένων με γκαουσιανό θόρυβο μπορεί να βοηθήσει σημαντικά τα score-based παραγωγικά μοντέλα.

Προκειμένου να επιτύχουμε τα οφέλη της διαταραχής με θόρυβο, χωρίς να αλλοιώσουμε την αρχική κατανομή των δεδομένων μας, χρησιμοποιούμε μια ακολουθία από διαταραχές με θόρυβο, οι οποίες συγκλίνουν στην πραγματική κατανομή δεδομένων. Αυτή η μέθοδος, εμπνευσμένη από την προσομοιωμένη ανόπτηση [19], επωφελείται από τις ενδιάμεσες κατανομές και μπορεί να βελτιώσει τα Langevin Dynamics.

Έστω $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$ μια ακολουθία L μειούμενων επιπέδων θορύβου. Διαταράσσουμε τα δεδομένα με γκαουσιανό θόρυβο $\mathcal{N}(\mathbf{0}, \sigma_i \mathbf{I})$, χρησιμοποιώντας όλα τα $\{\sigma_i\}_{i=1}^L$, με αποτέλεσμα μια ακολουθία διαταραγμένων κατανομών $q_{\sigma_i}(x) = \int p_{data}(t)\mathcal{N}(x, t, \sigma_i^2 \mathbf{I})dt$.

Πρέπει να εκτιμήσουμε το score κάθε θορυβοποιημένης κατανομής. Αυτό μπορεί να επιτευχθεί εκπαιδεύοντας ένα conditional score δίκτυο για όλες τις ενδιάμεσες κατανομές: $\forall \sigma_i \in \{\sigma_i\}_{i=1}^L : s_\theta(x, \sigma_i) \approx \nabla_x \log q_{\sigma_i}(x)$. Το $s_\theta(x, \sigma)$ καλείται *Noise Conditional Score Network (NCSN)* [4].

1.2.4 Εκπαίδευση NCSNs μέσω score matching

Τα NCSNs μπορούν να εκπαιδευτούν με denoising score matching, καθώς και με sliced score matching. Οι Song και Ermon [4] επιλέγουν το denoising score matching, καθώς είναι όχι μόνο ταχύτερο, αλλά και φυσικά κατάλληλο για το έργο της εκτίμησης των score κατανομών δεδομένων που έχουν διαταραχθεί από θόρυβο. Επιλέγουμε την κατανομή

θορύβου να είναι:

$$q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x, \sigma^2\mathbf{I}) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[\frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right] \tag{1.14}$$

Έτσι, το score θα είναι:

$$\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = \nabla_x\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right] = -\frac{\tilde{x}-x}{\sigma^2} \tag{1.15}$$

Ο στόχος του denoising score matching στην εξίσωση 1.11 γίνεται:

$$\ell(\theta, \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{data}(x)}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2\mathbf{I})}\left[\left\|s_\theta(\tilde{x}, \sigma) + \frac{\tilde{x}-x}{\sigma^2}\right\|_2^2\right] \tag{1.16}$$

Συνδυάζοντας την Εξ. 1.16 για όλα τα $\sigma \in \{\sigma_i\}_{i=1}^L$, λαμβάνουμε τον ενοποιημένο στόχο:

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^L) \triangleq \frac{1}{L}\sum_{i=1}^{L} \lambda(\sigma_i)\ell(\theta, \sigma_i) \tag{1.17}$$

όπου $\lambda(\sigma_i) > 0$ είναι ένας συντελεστής στάθμισης, που συχνά επιλέγεται ως $\lambda(\sigma_i) = \sigma_i^2$ [4].

1.3 Δειγματοληψία από NCSN μέσω annealed Langevin Dynamics

Μόλις εκπαιδευτεί το NCSN μας $s_\theta(x, \sigma)$, μπορούμε να χρησιμοποιήσουμε μια τροποποιημένη έκδοση των Langevin Dynamics, που ονομάζεται *annealed Langevin Dynamics*, για τη δειγματοληψία. Η διαδικασία ξεκινά με την αρχικοποίηση δειγμάτων από κάποια σταθερή εκ των προτέρων κατανομή. Μετά από κάθε επανάληψη, η μέθοδος μειώνει το επίπεδο θορύβου σ_i και το μέγεθος βήματος α_i. Ο πλήρης αλγόριθμος είναι ο 7.1.

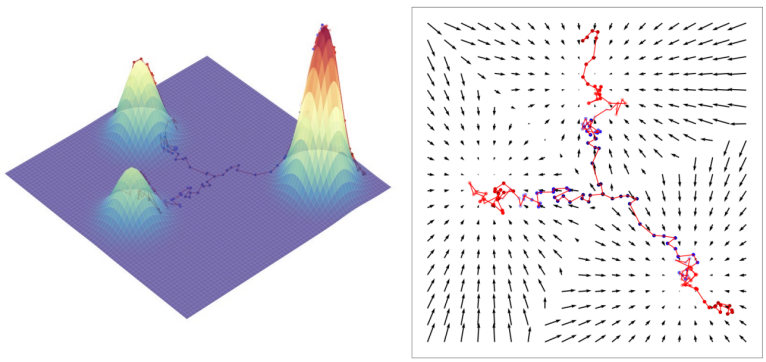

Figure 1.3. *Τυχαίες τροχιές δειγματοληψίας που παράγονται με Langevin dynamics [2].*

Στο Σχήμα 1.3 παρουσιάζονται τρεις τυχαίες τροχιές δειγματοληψίας που παράγονται με Langevin dynamics, για ένα μείγμα Γκαουσιανών και ξεκινούν από το ίδιο σημείο αρχικοποίησης. Το αριστερό υποδιάγραμμα δείχνει αυτές τις τροχιές σε 3D, ενώ το δεξιό υποδιάγραμμα τις δείχνει σε 2D, σε σχέση με την ground truth συνάρτηση score. Παρά το κοινό σημείο εκκίνησης, οι τροχιές παράγουν δείγματα από διαφορετικά modes, χάρη στις στοχαστικές διαταραχές θορύβου στα Langevin dynamics. Διαφορετικά, η διαδικασία θα ακολουθούσε πάντα ντετερμινιστικά το score στο ίδιο mode.

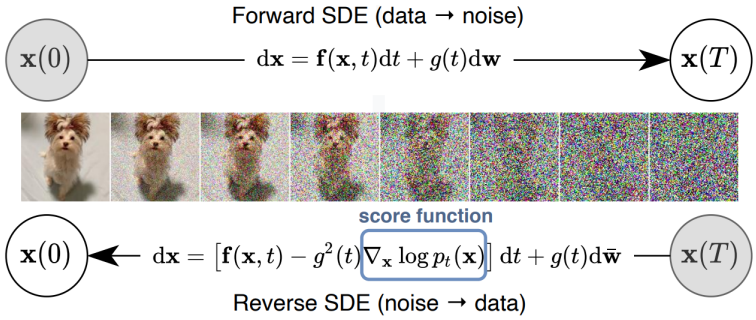

Figure 1.4. *Ευθεία και Αντίστροφη SDE [3].*

1.4 Στοχαστικές Διαφορικές Εξισώσεις

Στα προηγούμενα κεφάλαια, διερευνήσαμε τα μοντέλα διάχυσης διάχυσης και τα score-based παραγωγικά μοντέλα. Και οι δύο κατηγορίες μοντέλων βασίζονται στη σταδιακή διαταραχή των δεδομένων με αυξανόμενα επίπεδα θορύβου και στη μάθηση της αντιστροφής αυτής της διαδικασίας για τη δημιουργία νέων δειγμάτων από καθαρό θόρυβο, τα οποία ακολουθούν την αρχική κατανομή των δεδομένων. Τα Denoising Diffusion Probabilistic Models [6, 12] είναι πιθανοτικά μοντέλα, εκπαιδευμένα να αντιστρέφουν κάθε βήμα της διαδικασίας διαταραχής, ενώ τα Score-Based παραγωγικά μοντέλα [4] εκτιμούν τη συνάρτηση score των αρχικών δεδομένων, που έχουν διαταραχθεί σε διάφορες κλίμακες θορύβου, και χρησιμοποιούν Annealed Langevin Dynamics για τη δημιουργία νέων δειγμάτων δεδομένων. Οι Song *et al.* [3] απέδειξαν ότι και οι δύο κατηγορίες μοντέλων μπορούν να θεωρηθούν ως διακριτοποιήσεις στοχαστικών διαφορικών εξισώσεων, παρέχοντας έτσι μια ενοποιητική οπτική στις προηγούμενες προσεγγίσεις.

Το πλαίσιο των στοχαστικών διαφορικών εξισώσεων αποτελείται από έναν άπειρο αριθμό κατανομών, οι οποίες χρησιμοποιούνται για τη διάχυση των δεδομένων σε τυχαίο θόρυβο. Έτσι, η διαδικασία ευθείας διάχυσης δίνεται από μια προδιαγεγραμμένη στοχαστική διαφορική εξίσωση (SDE), η οποία είναι ανεξάρτητη από τα δεδομένα και δεν έχει εκπαιδεύσιμες παραμέτρους. Δεδομένου του score των πυκνοτήτων πιθανότητας ως συνάρτηση του χρόνου, και χρησιμοποιώντας την ευθεία SDE, μπορούμε να λάβουμε την SDE αντίστροφου χρόνου [20]. Η SDE αντίστροφου χρόνου μπορεί να προσεγγιστεί χρησιμοποιώντας ένα νευρωνικό δίκτυο που εξαρτάται από το χρόνο, για την εκτίμηση των score. Έτσι, μπορούμε να δημιουργήσουμε νέα δείγματα δεδομένων, χρησιμοποιώντας αριθμητικούς επιλυτές SDE. Οι ιδέες αυτές συνοψίζονται στο Σχήμα 1.4.

1.4.1 Θορυβοποίηση δεδομένων με χρήση SDEs

Οι προηγούμενες μέθοδοι βασίζονταν στη θορυβοποίηση των δεδομένων με διάφορα επίπεδα θορύβου. Επεκτείνουμε αυτή την ιδέα ενσωματώνοντας έναν άπειρο αριθμό επιπέδων θορύβου, με αποτέλεσμα να δημιουργούμε κατανομές θορυβοποιημένων δεδομένων που εξελίσσονται σύμφωνα με μια SDE, καθώς ο θόρυβος εντείνεται.

Η διαδικασία διάχυσης, την οποία εξετάσαμε σε προηγούμενα κεφάλαια, είναι η λύση

της ακόλουθης SDE:

$$dx = f(x,t)dt + g(t)dw \tag{1.18}$$

όπου w είναι η τυπική διαδικασία Wiener, $f(\cdot,t) : \mathbb{R}^D \to \mathbb{R}^D$ είναι μια διανυσματική συνάρτηση που ονομάζεται drift coefficient του $x(t)$ και $g(\cdot) : \mathbb{R} \to \mathbb{R}$ είναι ο diffusion coefficient του $x(t)$. Η λύση της SDE, είναι μια διαδικασία διάχυσης $\{x(t)\}$, όπου $t \in [0,T]$, τέτοια ώστε $x(0) \sim p_0$ (η κατανομή των δεδομένων) και $x(T) \sim p_T$ (η προηγούμενη κατανομή).

Παρατηρούμε μια αναλογία μεταξύ των διακριτών βημάτων της διάχυσης, που συζητήθηκαν σε προηγούμενα κεφάλαια, και της συνεχούς διάχυσης που περιγράφεται από την SDE. Συγκεκριμένα, τα διακριτά χρονικά βήματα $i = 1, 2, \ldots, L$ γενικεύονται σε χρονικά βήματα $t \in [0,T]$. Επιπλέον, τα διακριτά επίπεδα θορύβου $\sigma_1, \sigma_2, \ldots, \sigma_L$, τα οποία επιλέχθηκαν χειροκίνητα στη διακριτή περίπτωση, αντιστοιχούν στην SDE στη συνεχή περίπτωση, η οποία είναι επίσης χειροκίνητα σχεδιασμένη. Με άλλα λόγια, ο τρόπος σχεδιασμού της SDE σχετίζεται άμεσα με τα επίπεδα θορύβου που επιλέγουμε για τη διαταραχή στη διακριτή περίπτωση. Παρόμοια με το πώς στη διακριτή περίπτωση, τα επίπεδα θορύβου είναι χαρακτηριστικά του μοντέλου, στη συνεχή περίπτωση, η SDE είναι χαρακτηριστική του μοντέλου.

Στην επόμενη υποενότητα, διερευνούμε τις διαταραχές θορύβου που χρησιμοποιούνται στα Score-Based Generative Models και στα Denoising Diffusion Probabilistic Models, τα οποία μπορούν να θεωρηθούν ως διακριτοποιήσεις των SDEs *Variance Exploding (VE)* και *Variance Preserving (VP)*, αντίστοιχα.

1.5 Variance Exploding και Variance Preserving SDEs

Οι διαταραχές θορύβου που χρησιμοποιούνται στα Score-Based Generative Models και στα Denoising Diffusion Probabilistic Models είναι διακριτοποιήσεις δύο διαφορετικών τύπων SDEs [3].

Στην περίπτωση των Score-Based Generative Models, κάθε πυρήνας διαταραχής $p_{\sigma_i}(x|x_0)$ αντιστοιχεί στην κατανομή του x_i, στην ακόλουθη αλυσίδα Markov:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, z_{i-1}, \quad i = 1, \ldots, N \tag{1.19}$$

όπου $z_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Θεωρούμε άπειρες κλίμακες θορύβου ($N \to \infty$). Στην περίπτωση αυτή, η $\{\sigma_i\}_{i=1}^N$ γίνεται συνάρτηση $\sigma(t)$, η κατανομή θορύβου z_i γίνεται $z(t)$ και η αλυσίδα Markov $\{x_i\}_{i=1}^N$ γίνεται συνεχής στοχαστική διαδικασία. Χρησιμοποιούμε τον συμβολισμό $\{x(t)\}_{t=0}^1$, όπου t είναι μια συνεχής χρονική μεταβλητή $t \in [0,1]$. Η αντίστοιχη SDE, της οποίας η λύση είναι $\{x(t)\}_{t=0}^1$, είναι:

$$dx = \sqrt{\frac{d\left[\sigma^2(t)\right]}{dt}}\, dw \tag{1.20}$$

Για Denoising Diffusion Probabilistic Models, με πυρήνες διαταραχών $\{p_{\alpha_i}(x|x_0)\}_{i=1}^N$, η διακριτή αλυσίδα Markov είναι:

$$x_i = \sqrt{1-\beta_i}\, x_{i-1} + \sqrt{\beta_i}\, z_{i-1}, \quad i = 1, \ldots, N \tag{1.21}$$

Όταν $N \to \infty$, η Εξ. 1.21 συγκλίνει στην ακόλουθη SDE:

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw \tag{1.22}$$

Οι διαταραχές θορύβου που χρησιμοποιούνται στα Score-Based Generative Models είναι μια διακριτοποίηση της Εξ. 1.20 και οι διαταραχές θορύβου που χρησιμοποιούνται στα Denoising Diffusion Probabilistic Models είναι μια διακριτοποίηση της Εξ. 1.22.

Δεδομένου ότι η λύση της Εξ. 1.20 είναι πάντα μια διαδικασία εκρηγνυόμενης διακύμανσης, καθώς $t \to \infty$, η εξίσωση είναι γνωστή ως *Variance Exploding (VE)* SDE. Ομοίως, η λύση της εξίσωσης 1.22 είναι πάντα μια διαδικασία σταθερής διακύμανσης της μονάδας, εφόσον η αρχική κατανομή έχει μοναδιαία διακύμανση [3]. Συνεπώς, η Εξ. 1.22 είναι γνωστή ως *Variance Preserving (VP)* SDE.

1.5.1 Αντιστροφή της SDE για την παραγωγή δειγμάτων

Εξετάσαμε πώς στην διακριτή περίπτωση, μπορούμε να αντιστρέψουμε τη διαδικασία ευθείας διάχυσης για να παράξουμε νέα δείγματα, ξεκινώντας από προηγούμενες κατανομές καθαρού θορύβου. Ομοίως, στη συνεχή περίπτωση, ξεκινώντας με ένα δείγμα $x(T) \sim p_T$ και αντιστρέφοντας τη διαδικασία της SDE, μπορούμε να παράγουμε νέα δείγματα $x(0) \sim p_0$. Ο Anderson [20] απέδειξε ότι η αντιστροφή μιας διαδικασίας διάχυσης οδηγεί σε μια άλλη διαδικασία διάχυσης, η οποία τρέχει προς τα πίσω στο χρόνο και δίνεται από την αντίστροφη SDE:

$$dx = \left[f(x,t) - g(t)^2\nabla_x \log p_t(x)\right]dt + g(t)d\bar{w} \tag{1.23}$$

Δεδομένου ότι αυτή η SDE τρέχει προς τα πίσω στο χρόνο, το dt αντιπροσωπεύει ένα απειροελάχιστο αρνητικό χρονικό βήμα και το $\bar{w}$ είναι η τυπική διαδικασία Wiener, καθώς ο χρόνος ρέει από το T στο 0. Για να πάρουμε δείγμα από το p_0, πρέπει να προσομοιώσουμε την αντίστροφη SDE, η οποία απαιτεί την εκτίμηση της συνάρτησης score $\nabla_x \log p_t(x)$. Σημειώστε ότι η συνάρτηση score ταυτίζεται με αυτή που μαθαίνουν τα score-based μοντέλα.

Αφού εκπαιδευτεί ένα score-based μοντέλο για την εκτίμηση της συνάρτησης score, μπορούμε να το χρησιμοποιήσουμε για την προσομοίωση της αντίστροφης SDE και την επίλυσή της, για να δημιουργήσουμε δείγματα που ταιριάζουν με την αρχική κατανομή δεδομένων, δηλαδή p_0. Εξετάζουμε δύο προσεγγίσεις στην επίλυση της SDE, αριθμητικούς επιλυτές SDE γενικού σκοπού και δειγματολήπτες Predictor-Corrector (PC).

Αριθμητικοί επιλυτές SDE γενικού σκοπού Η αντίστροφη SDE μπορεί να επιλυθεί με τη χρήση αριθμητικών επιλυτών SDE, οι οποίοι είναι ικανοί να προσεγγίζουν τροχιές από SDEs. Οποιοσδήποτε επιλυτής SDE μπορεί να χρησιμοποιηθεί για την παραγωγή δειγμάτων. Οι πιο δημοφιλείς περιλαμβάνουν τις μεθόδους Euler-Maruyama και τις στοχαστικές μεθόδους Runge-Kutta.

Δειγματολήπτες πρόβλεψης-διόρθωσης Μπορούμε να αξιοποιήσουμε το score-based μοντέλο για να βελτιώσουμε τις λύσεις της αντίστροφης SDE. Οι δειγματολήπτες Predictor-Corrector αποτελούνται από δύο κύρια στοιχεία. Το πρώτο είναι ένας προβλεπτής (Predictor), ο οποίος κάνει μία εκτίμηση για το δείγμα, ενώ το δεύτερο είναι ένας διορθωτής (Corrector), ο οποίος διορθώνει την κατανομή του προβλεπόμενου δείγματος. Αυτοί οι δειγματολήπτες

είναι επαναληπτικοί, παράγωντας τόσο μια πρόβλεψη όσο και μια διόρθωση, σε κάθε χρονικό βήμα. Ο προβλεπτής είναι ένας αριθμητικός επιλυτής SDE, ενώ ο διορθωτής είναι μια score-based προσέγγιση MCMC, όπως η Langevin MCMC.

Chapter 2

Καθοδήγηση από Discriminator

Σε μια προσέγγιση που είναι εμπνευσμένη από τα Generative Adversarial Networks, οι Kim *et al.* [5] τροποποιούν τη διαδικασία δειγματοληψίας διάχυσης χρησιμοποιώντας ένα δίκτυο διαχωριστή (discriminator) για να γεφυρώσουν το χάσμα μεταξύ του score του μοντέλου και του πραγματικού score των δεδομένων. Στη μέθοδό τους, Καθοδήγηση Διαχωριστή (*Discriminator Guidance*), διατηρούν το προ-εκπαιδευμένο μοντέλο score ως σταθερό στοιχείο και εισάγουν ένα δίκτυο discriminator για την ταξινόμηση πραγματικών και παραγόμενων δεδομένων, σε διάφορες κλίμακες θορύβου. Η μέθοδος ενσωματώνει έναν όρο διόρθωσης στο μοντέλο score, καθοδηγούμενο από την ανατροφοδότηση του discriminator, ο οποίος συμβάλλει στην καθοδήγηση της παραγωγής δειγμάτων προς πιο ρεαλιστικές διαδρομές.

2.1 Διόρθωση του score του μοντέλου

Μετά την εκπαίδευση των score, παράγουμε δείγματα χρησιμοποιώντας την παραγωγική διαδικασία που περιγράφεται από την αντίστροφη SDE:

$$dx = \left[f(x,t) - g(t)^2 s_{\theta_\infty}(x,t)\right] dt + g(t) d\bar{w} \tag{2.1}$$

όπου s_{θ_∞} συμβολίζει το προ-εκπαιδευμένο δίκτυο score. Εάν το τοπικό βέλτιστο θ_∞ αποκλίνει από το ολικό βέλτιστο θ_*, η παραγωγική διαδικασία μπορεί να διαφέρει από την πραγματική διαδικασία αντίστροφου χρόνου. Το θεώρημα 9.1 αποδεικνύει ότι η παραγωγική διαδικασία της Εξ. 2.1 μπορεί να ευθυγραμμιστεί με την πραγματική διαδικασία της Εξ. 1.23 προσαρμόζοντας το score του μοντέλου. Η διαφορά μεταξύ των δύο διαδικασιών γεφυρώνεται από τον διορθωτικό όρο, ο οποίος είναι μη μηδενικός όποτε $\theta_\infty \neq \theta_*$.

2.2 Καθοδήγηση από Discriminator

Ο διορθωτικός όρος $c_{\theta_\infty}(x,t) = \nabla \log \frac{p_r^t(x)}{p_{\theta_\infty}^t(x)}$ είναι γενικά απροσδιόριστος λόγω του απροσδιόριστου λόγου πυκνότητας $\frac{p_r^t}{p_{\theta_\infty}^t}$. Για να το αντιμετωπίσουν αυτό, οι Kim *et al.* [5] προσεγγίζουν τον λόγο πυκνότητας εκπαιδεύοντας έναν discriminator σε κάθε επίπεδο θορύβου t. Για την εκπαίδευση του discriminator, δημιουργούνται ψεύτικα δείγματα από τη διαδικασία που περιγράφεται στην Εξ. 2.1, ίσα σε αριθμό με τα πραγματικά δείγματα δεδομένων. Τα

πραγματικά και τα ψεύτικα δεδομένα ταξινομούνται στη συνέχεια με τη χρήση της δυαδικής σταυροειδούς εντροπίας (BCE):

$$\begin{aligned}\mathcal{L}_\phi = \int \lambda(t)(&\mathbb{E}_{p_r^t(x)}[-\log d_\phi(x,t)] \\ &+ \mathbb{E}_{p_{\partial_\infty}^t(x)}[-\log(1-d_\phi(x,t))])dt,\end{aligned} \tag{2.2}$$

όπου λ αντιπροσωπεύει το χρονικό βάρος.

Εκφρασμένος ως προς το βέλτιστο διακριτικό ϕ_* που προκύπτει από το $\mathcal{L}_\phi$, ο διορθωτικός όρος είναι:

$$c_{\partial_\infty}(x,t) = \nabla \log \frac{d_{\phi_*}(x,t)}{1-d_{\phi_*}(x,t)}.$$

Για την εκτίμηση του διορθωτικού όρου c_{∂_∞}, χρησιμοποιούμε έναν νευρωνικό discriminator ϕ:

$$c_{\partial_\infty}(x,t) \approx c_\phi(x,t) = \nabla \log \frac{d_\phi(x,t)}{1-d_\phi(x,t)}.$$

Με αυτή την εύχρηστη εκτίμηση του όρου διόρθωσης, η καθοδήγηση με discriminator (DG) ορίζεται ως εξής:

$$dx = [f(x,t) - g^2(t)(s_{\partial_\infty} + c_\phi)(x,t)]dt + g(t)d\bar{w}. \tag{2.3}$$

Chapter 3

Exposure Bias

Η επαναληπτική αλυσίδα δειγματοληψίας των μοντέλων διάχυσης συχνά εκτείνεται σε χιλιάδες βήματα λόγω της γκαουσιανής παραδοχής της αντίστροφης διάχυσης, η οποία ισχύει για μικρά μεγέθη βημάτων [6]. Ωστόσο, αυτό οδηγεί στο φαινόμενο της μεροληψίας έκθεσης (exposure bias), όπως καταδεικνύεται από τους Ning *et al.* [21]. Το exposure bias προκύπτει από τη διαφορά μεταξύ των δεδομένων εισόδου κατά την εκπαίδευση (x_t) και την δειγματοληψία ($\hat{x}t$). Αυτή η διαφορά οδηγεί σε διαφορετικές προβλέψεις θορύβου, $\epsilon_\theta(x_t)$ και $\epsilon_\theta(\hat{x}_t)$, οδηγώντας σε συσσώρευση σφάλματος γνωστή ως ολίσθηση δειγματοληψίας (sampling drift) [22]. Οι Ning *et al.* [23] προτείνουν την Κλιμάκωση Έψιλον (*Epsilon Scaling*), μια μέθοδο που ενσωματώνεται απευθείας στη διαδικασία δειγματοληψίας, χωρίς να απαιτείται επανεκπαίδευση του μοντέλου.

3.1 Σφάλμα πρόβλεψης και exposure bias

Οι Ning *et al.* [21] εντοπίζουν τη συσσώρευση σφάλματος μεταξύ των βημάτων inference λόγω της ασυμφωνίας μεταξύ των σταδίων εκπαίδευσης και δειγματοληψίας. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο μαθαίνει με το ground truth ζεύγος (x_t, x_{t-1}), αλλά κατά τη διάρκεια της δειγματοληψίας, βασίζεται στο παραγόμενο $\hat{x}_t$, προκαλώντας πιθανή συσσώρευση σφαλμάτων. Αυτή η ασυμφωνία οδηγεί στο exposure bias που παρατηρείται σε άλλα παραγωγικά μοντέλα [24, 25]. Η κατανομή εκπαίδευσης είναι $q(x_t|x_0)$, ενώ η κατανομή δειγματοληψίας είναι $q(\hat{x}_t|x_{t+1}, x_\theta^{t+1})$. Αυτό οδηγεί στο πρόβλημα του exposure bias, καθώς το $\epsilon_\theta(x_t)$ και το $\epsilon_\theta(\hat{x}_t)$ διαφέρουν.

Οι Xiao *et al.* [26] παραμετροποιούν το $p_\theta(x_{t-1}|x_t) = q(x_{t-1}|x_t, x_\theta^t)$, όπου x_θ^t είναι το προβλεπόμενο x_0. Ωστόσο, τα σφάλματα πρόβλεψης κατά την εκτίμηση του x_0 οδηγούν σε $q(x_{t-1}|x_t, x_0) \neq q(x_{t-1}|x_t, x_\theta^t)$, προκαλώντας το exposure bias. Οι Ning *et al.* [23] ποσοτικοποιούν αναλυτικά αυτή την ασυμφωνία στα DDPMs, αποκαλύπτοντας ότι η διακύμανση της κατανομής δειγματοληψίας υπερβαίνει τη διακύμανση της κατανομής εκπαίδευσης, οδηγώντας σε συσσώρευση σφαλμάτων.

3.2 Related Work

Για την αντιμετώπιση της μεροληψίας έκθεσης, οι Ning *et al.* [21] προτείνουν τη ρητή μοντελοποίηση του σφάλματος πρόβλεψης κατά τη διάρκεια της εκπαίδευσης, προσομοιώνοντας

την ασυμφωνία παρέχοντας μια πιο θορυβώδη έκδοση του x_t. Οι Li *et al.* [22] μετατοπίζουν το χρονικό βήμα δειγματοληψίας t με βάση τη διακύμανση του παραγόμενου δείγματος, μετριάζοντας αποτελεσματικά το exposure bias χωρίς επανεκπαίδευση. Ωστόσο, ο προσδιορισμός της μετατόπισης του χρονικού βήματος είναι δύσκολος.

3.3 Epsilon Scaling

Οι Ning *et al.* [23] εισάγουν το *Epsilon Scaling*, μειώνοντας τη μεροληψία έκθεσης με την κλιμάκωση του προβλεπόμενου παράγοντα θορύβου ϵ_θ^s κατά τη διάρκεια της δειγματοληψίας. Με τη μείωση του υπερεκτιμώμενου μεγέθους του ϵ_θ^s, η μεροληψία έκθεσης μετριάζεται χωρίς επανεκπαίδευση. Η μέθοδός τους βασίζεται στην παρατήρηση ότι τόσο το ϵ_θ^s όσο και το ϵ_θ^t προέρχονται από την ίδια είσοδο, $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, αλλά αποκλίνουν λόγω σφαλμάτων στην αλυσίδα δειγματοληψίας. Η μέθοδος δεν προσθέτει υπολογιστική επιβάρυνση και είναι μια plug-in λύση.

Chapter 4

Μέθοδος και Αποτελέσματα

4.1 Ποσοτικοποιώντας το Exposure Bias

Κατά την εκπαίδευση, το δίκτυο λαμβάνει είσοδο x_t, ενώ κατά τη δειγματοληψία, λαμβάνει είσοδο $\hat{x}_t$ (Πίνακας 10.1). Αυτή η ασυμφωνία έχει ως αποτέλεσμα την απόκλιση μεταξύ των προβλέψεων θορύβου κατά τη διάρκεια της εκπαίδευσης, ϵ^t_θ, και κατά τη διάρκεια της δειγματοληψίας, ϵ^s_θ. Ακολουθώντας τους Ning *et al.* [23], μετράμε το sampling drift ως τη διαφορά μεταξύ ϵ^t_θ και ϵ^s_θ. Δεδομένου ότι το ground truth του ϵ^s_θ είναι μη προσβάσιμο κατά τη δειγματοληψία, χρησιμοποιούμε την $L2$-νόρμα για να ποσοτικοποιήσουμε το exposure bias [23].

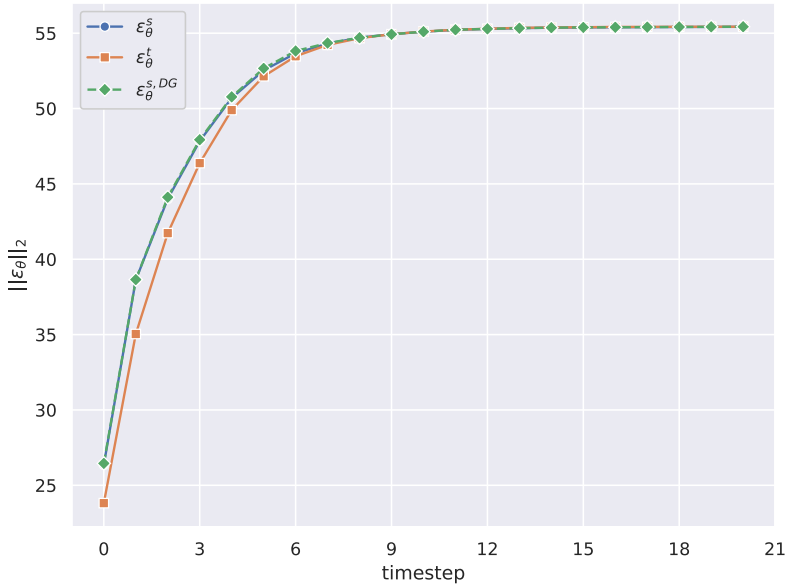

Figure 4.1. *Μοντέλο EDM, επιλύτης Euler 1ης τάξης. L2-νόρμα του $\epsilon_\theta(\cdot)$ κατά τη διάρκεια δειγματοληψίας 21 βημάτων, δειγματοληψίας με DG και εκπαίδευσης. Η L2-νόρμα υπολογίζεται χρησιμοποιώντας 50k δείγματα σε κάθε χρονικό βήμα. Η δειγματοληψία γίνεται από τα δεξιά προς τα αριστερά.*

Στα Σχήματα 4.1 και 4.2, απεικονίζουμε την $L2$-νόρμα των ϵ^t_θ, ϵ^s_θ και $\epsilon^{s,DG}_\theta$ χρησιμοποιώντας τους επιλύτες ODE Euler και Heun. $\epsilon^{s,DG}_\theta$ είναι η πρόβλεψη θορύβου στο μοντέλο EDM-

G++. Για τον επιλυτή Euler, το $\epsilon_\partial^{s,DG}$ έχει μεγαλύτερη $L2$-νόρμα από το ϵ_∂^t, που σχεδόν ταιριάζει με το ϵ_∂^s. Με τον επιλυτή Heun, η διαφορά είναι μικρότερη, αλλά το $\epsilon_\partial^{s,DG}$ έχει και πάλι μεγαλύτερη $L2$-νόρμα, υποδεικνύοντας ότι η καθοδήγηση του discriminator δεν εξαλείφει το exposure bias- τα σφάλματα πρόβλεψης συσσωρεύονται.

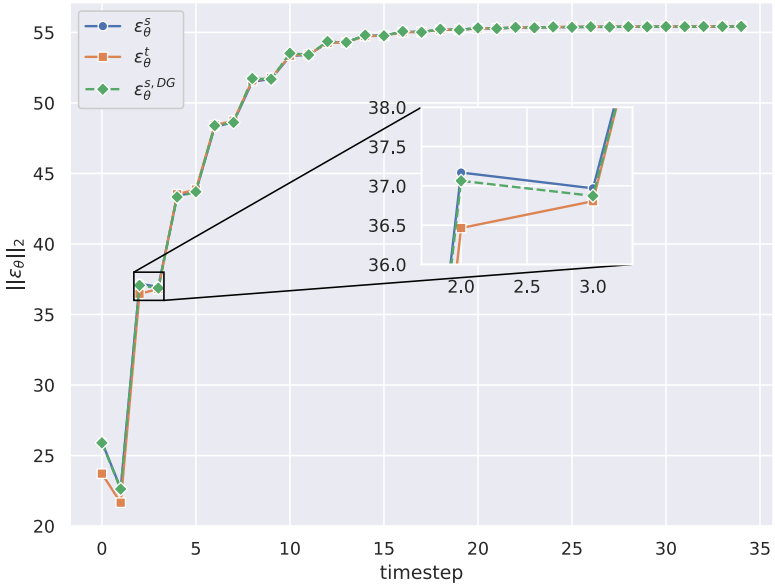

Figure 4.2. *Μοντέλο EDM, επιλύτης Heun 2ης τάξης. L2-νόρμα του $\epsilon_\partial(\cdot)$ κατά τη διάρκεια δειγματοληψίας 35 βημάτων, δειγματοληψίας με DG και εκπαίδευσης. Η L2-νόρμα υπολογίζεται χρησιμοποιώντας 50k δείγματα σε κάθε χρονικό βήμα. Η δειγματοληψία γίνεται από τα δεξιά προς τα αριστερά.*

4.2 Προτεινόμενο Framework

Χρησιμοποιούμε το μοντέλο EDM των Karras *et al.* ως εκτιμητή score λόγω του λεπτομερούς σχεδιασμού του, που επιτυγχάνει σημαντική βελτίωση της ποιότητας δειγμάτων (FID score 1.97 στο CIFAR-10). Δεδομένης της συσχέτισης του exposure bias με το FID score [23], υποθέτουμε ότι το EDM έχει μειωμένο exposure bias.

Για τον discriminator, ακολουθώντας τους Kim *et al.* [5], χρησιμοποιούμε δύο κωδικοποιητές U-Net, με τον πρώτο παγωμένο και τον δεύτερο fine-tuned.

Στην ενότητα 10.2, συζητάμε μεθόδους για τη μείωση της μεροληψίας έκθεσης. Επιλέγουμε το Epsilon Scaling [23], μια plug-in μέθοδο, χωρίς εκπαίδευση, που έχει αποδειχθεί αποτελεσματική στη μείωση του Exposure Bias και στη βελτίωση των αποτελεσμάτων (FID score). Παρουσιάζουμε το Epsilon Scaling στην ενότητα 10.3.

Αν και το EDM εξάγει τη συνάρτηση score s_∂ και όχι το ϵ, το ϵ μπορεί να εξαχθεί σε κάθε βήμα δειγματοληψίας και να χρησιμοποιηθεί για το Epsilon Scaling [23].

Για το πρόγραμμα scaling λ_t, οι Ning *et al.* [23] προτείνουν μια γραμμική συνάρτηση $\lambda_t = kt + b$ με σταθερές k και b. Προτείνουν ένα ομοιόμορφο πρόγραμμα λ_t ($k = 0$) για λόγους πρακτικότητας. Τα πειράματά μας επιβεβαιώνουν παρόμοια ή υπο-βέλτιστα

αποτελέσματα με γραμμικό λ_t σε σύγκριση με το ομοιόμορφο πρόγραμμα $\lambda_t = b$. Έτσι προχωράμε σε περαιτέρω διερεύνηση του σταθερού παράγοντα κλιμάκωσης.

4.3 Αποτελέσματα

4.3.1 Euler Solver

Σε αυτή την ενότητα, παρουσιάζουμε αποτελέσματα με τη χρήση του επιλυτή ODE 1ης τάξης Euler. Για τη μέθοδο Discriminator Guidance (Kim *et al.* [5]), σημειώνεται μια βιβλιογραφική παράλειψη στην αναφορά των επιδόσεων του μοντέλου EDM-G++ με χρήση του επιλυτή ODE Euler. Οι βαθμολογίες FID του EDM-G++ στο σύνολο δεδομένων CIFAR-10 υπολογίζονται με διαφορετικά χρονικά βήματα και βάρος Discriminator Guidance (w^{DG}) και παρουσιάζονται στο Σχήμα 4.3.

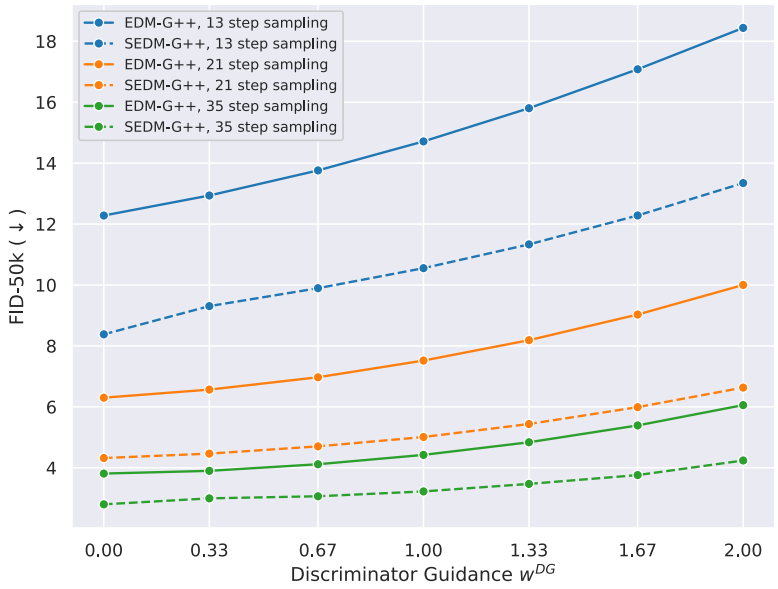

Figure 4.3. *Μελέτη FID-50k σε σχέση με βάρος DG και παράγοντα κλιμάκωσης (Euler Solver).*

Η ποιότητα του EDM-G++ μειώνεται με την αύξηση του w^{DG} με τον Euler solver. Το μειωμένο FID score αποδίδεται στην απουσία διορθωτικού βήματος στον επιλύτη 1ης τάξης, αναδεικνύοντας τη σημασία του διορθωτικού βήματος για την ποιότητα δειγμάτων (Σχ. 4.3).

Η μέθοδος epsilon scaling αποδεικνύεται αποτελεσματική για τα μοντέλα διάχυσης που καθοδηγούνται από discriminator. Το SEDM-G++ μειώνει τις βαθμολογίες FID σε διάφορα πλήθη χρονικών βημάτων και τιμές w^{DG} χρησιμοποιώντας τον επιλύτη Euler. Τα τυχαία δείγματα παρουσιάζουν ποιοτικές βελτιώσεις στην ευκρίνεια και τον ορισμό του σχήματος (Εικ. 4.4).

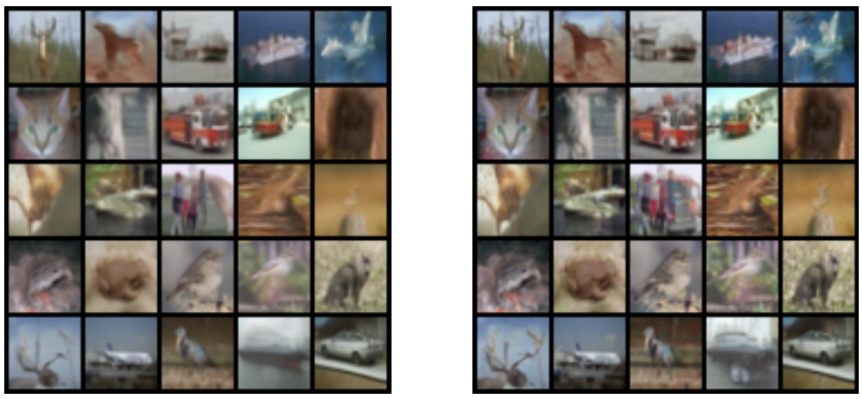

Figure 4.4. *Τυχαία δείγματα από το EDM-G++ (αριστερά) και το SEDM-G++ (δεξιά).*

4.3.2 Heun Solver

Διερευνάται η απόδοση του SEDM-G++ με τη χρήση του επιλυτή ODE 2ης τάξης Heun. Διεξάγεται μελέτη για το βάρος του discriminator ($w^{DG}_{t,1st}$) και τον παράγοντα κλιμάκωσης έψιλον ($\lambda_t = b$), ως προς το FID-10k (Σχήμα 4.5). Προσδιορίζονται οι βέλτιστες τιμές $w^{DG}_{t,1st}$ και ακολουθεί ολοκληρωμένη μελέτη πάνω σε αυτά, ως προς το FID-50k (Σχ. 4.6).

Model	NFE(↓)	FID(↓)
VDM (Kingma *et al.*, 2021 [27])	1000	7.41
DDPM (Ho *et al.*, 2020 [12])	1000	3.17
iDDPM (Nichol & Dhariwal, 2021 [28])	1000	2.90
Soft Truncation (Kim *et al.*, 2022 [29])	2000	2.47
INDM (Kim *et al.*, 2022 [30])	2000	2.28
CLD-SGM (Dockhorn *et al.*, 2022 [31])	312	2.25
NCSN++ (Song *et al.*, 2020 [3])	2000	2.20
LSGM (Vahdat *et al.*, 2021 [32])	138	2.10
EDM (Karras *et al.*, 2022 [14])	**35**	1.97
EDM-G++ (Kim *et al.*, 2023 [5])	**35**	1.77
SEDM-G++ (ours)	**35**	**1.73**

Table 4.1. *Σύγκριση FID-50k στην σύνθεση εικόνας στο unconditional CIFAR-10.*

Το SEDM-G++ ξεπερνά τα υπάρχοντα μοντέλα, επιτυγχάνοντας state-of-the-art FID score 1.73, με τις βέλτιστες υπερπαραμέτρους, στην σύνθεση εικόνων από το unconditional CIFAR-10. Η σύγκριση των επιδόσεων στον πίνακα 4.1 δείχνει την υπεροχή του SEDM-G++. Αναφέρουμε ότι το κέρδος FID μέσω του Epsilon Scaling στον επιλυτή Euler είναι πιο έντονο σε σύγκριση με τον επιλυτή Heun, γεγονός που αποδίδεται σε μικρότερη συσσώρευση σφαλμάτων, σε επιλύτες ODE υψηλότερης τάξης. Τα αποτελέσματα αναδεικνύουν την υπεροχή του Heun επιλυτή στα μοντέλα διάχυσης, η οποία αποδίδεται στο χαμηλότερο σφάλμα πρόβλεψης και στα βήματα διόρθωσης που μετριάζουν το exposure bias [23].

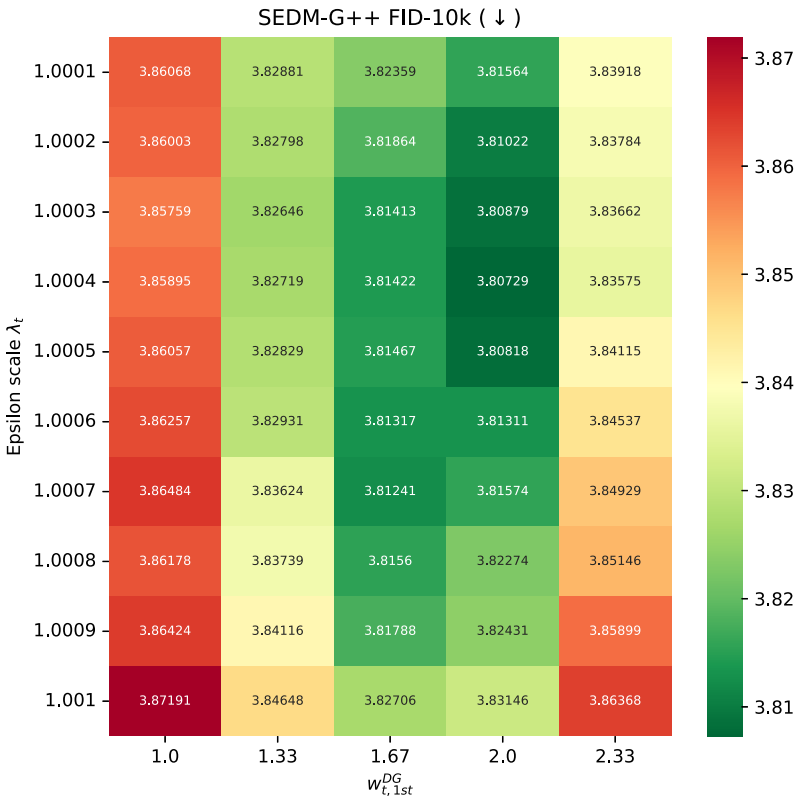

Figure 4.5. *Μελέτη FID-10k για το βάρος του DG και τον παράγοντα κλιμάκωσης (Heun Solver).*

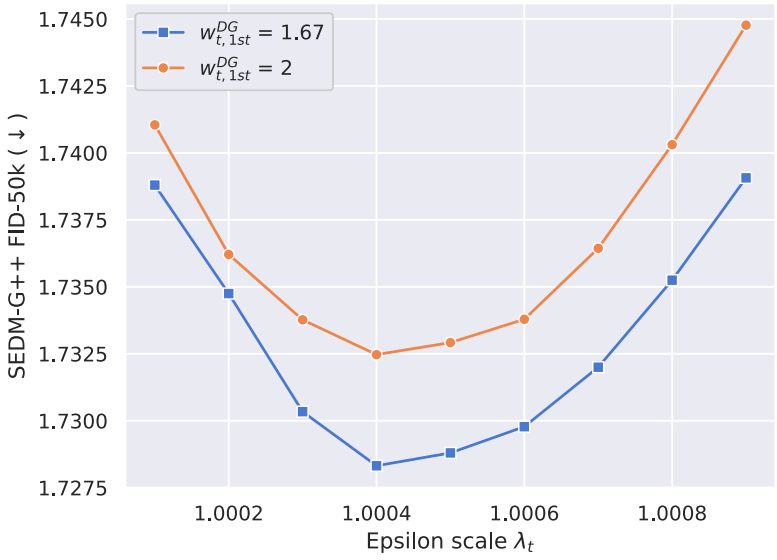

Figure 4.6. *Μελέτη FID-50k για τα βάρη του DG με τις καλύτερες επιδόσεις (Heun Solver).*

Part II

English Version

Chapter 5

Introduction

Diffusion models, initially proposed by Sohl-Dickstein *et al.* in 2015 [6], have excelled in various domains, including image generation, audio generation [7, 8], and video generation [9, 10, 11]. In the domain of image synthesis, diffusion models have seen significant advancements in subsequent years through the work of Song and Ermon (2019) [4], Ho *et al.* (2020) [12], and Nichol and Dhariwal (2021) [13]. In 2021, Song *et al.* presented a novel approach that unifies score-based models and Denoising Diffusion Probabilistic Models (DDPMs) by employing stochastic differential equations (SDEs) [3]. Moreover, Karras *et al.* (2022) presented a comprehensive exploration of the diffusion model design space and introduced the EDM model [14], which implemented a range of optimizations to both the sampling and training processes, leading to a substantial enhancement in performance and sample quality.

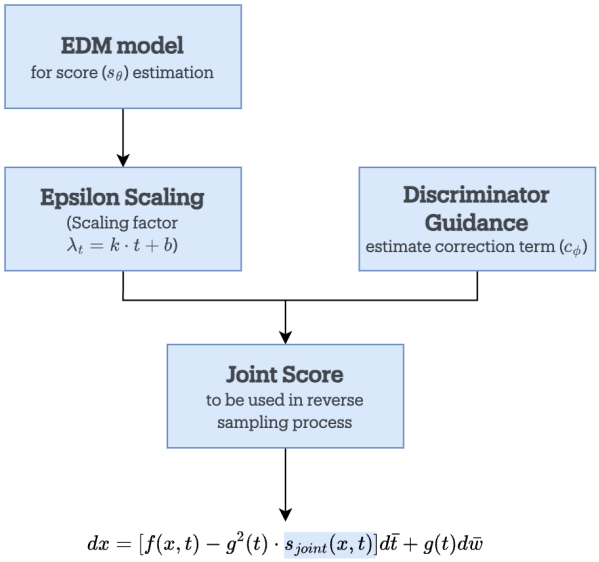

Figure 5.1. *Overview of our proposed SEDM-G++.*

Diffusion models [6] have demonstrated exceptional performance across diverse domains such as image, audio, and video generation [7, 8, 9, 10, 11]. In the realm of image synthesis, substantial progress has been made in recent years through various contributions [1, 3, 4, 12, 13, 14, 33, 34, 35, 36]. Diffusion models find a wide range of applica-

tions, including text to image generation [37, 38, 39], image inpainting [33, 40, 41, 42], image editing [41, 43, 44, 45, 46, 47, 48], super-resolution [41, 49], point cloud generation [50], 3D shape generation [51] and vision decoding [52].

The computational cost associated with training new score models from the ground up has initiated research endeavors which employ pre-existing score models and enhance the quality of generated samples via refinements in the sampling procedure.

In an effort loosely inspired by Generative Adversarial Networks (GANs), Kim *et al.* [5] modify the diffusion sampling process by utilizing a discriminator network to bridge the gap between the model score and the true data score. In their method, *Discriminator Guidance* (DG), they maintain the pre-trained score model as a fixed component and introduce a discriminator network to classify real and generated data across different noise scales. The method incorporates a correction term into the model score, guided by the discriminator's feedback, which helps steer the sample generation towards more realistic paths.

The iterative sampling chain in diffusion models is long, usually requiring thousands of steps due to the Gaussian assumption of reverse diffusion, which only holds for small step sizes [6]. This leads to the exposure bias problem, illustrated by Ning *et al.* [21]. Exposure bias refers to the discrepancy between the input data during training and inference phases. During training, the model is consistently exposed to the ground truth training sample x_t. However, during inference, the model relies on the previously generated sample, $\hat{x}_t$. This distinction between x_t and $\hat{x}_t$ results in a difference between $\epsilon_\theta(x_t)$ and $\epsilon_\theta(\hat{x}_t)$, where ϵ_θ is the model's noise prediction. This disparity between the two predictions results in error accumulation and deviations in the sampling process, known as "sampling drift" [22]. Ning *et al.* [23] propose an effective method, *Epsilon Scaling*, for alleviating exposure bias, which is incorporated directly into the sampling process and requires no training or fine-tuning of the model.

The preceding research prompts us to inquire whether Discriminator Guidance is effective in mitigating the accumulation of exposure bias in the sampling process. Our findings indicate that, despite notable enhancements in sample quality, Discriminator Guidance is ineffective in alleviating exposure bias in Diffusion Models. We propose SEDM-G++, which incorporates a modified sampling approach, combining Discriminator Guidance and Epsilon Scaling. We test our method on top of the pre-trained EDM model [14] and show that the proposed sampling process achieves improved sample quality, while reducing exposure bias.

Our contributions can be summarized as follows.

- We investigate exposure bias in discriminator guided diffusion models.
- We propose SEDM-G++, which incorporates a sampling approach combining Discriminator Guidance and Epsilon Scaling.
- Our proposed method improves sample quality across the board and outperforms the current state-of-the-art, by achieving an FID score of 1.73 on the unconditional CIFAR-10 dataset.

Chapter 6

Diffusion Models

Diffusion models were first introduced by Sohl-Dickstein *et al.* in 2015 [6]. Inspired by non-equilibrium thermodynamics, diffusion models gradually perturb data with noise, in an iterative forward diffusion process. The aim is to learn a reverse diffusion process, which removes the addition of noise. Thus, the model becomes capable of generating realistic samples from noise.

6.1 Inspiration behind diffusion models

The theoretical underpinning of diffusion models resides in the notion that diffusion processes can be strategically harnessed to disassemble the structural components inherent in our data distribution. To elucidate this theoretical framework in a more concrete manner, one may picture a hypothetical scenario involving a container filled with water into which a small quantity of coloured dye is introduced. If we assume that the concentration of dye molecules can be regarded as a representation of a probability distribution, the central aim of a generative model is to acquire an in-depth comprehension of this probabilistic structure. It is imperative to acknowledge that this endeavour is typically quite challenging.

Nevertheless, even in situations where we cannot directly construct a model to elucidate the intrinsic structure of our data distribution, there exists a viable alternative strategy. This involves the transformation of our data distribution into a substantially simplified distribution that can be readily modelled.

In the context of this physical analogy, if one permits the diffusion process to evolve over a sufficiently protracted period, the dye molecules will eventually disperse uniformly throughout the container, thereby yielding a state of equilibrium characterized by a uniform distribution. Initially, the usefulness of this transformation may not be readily apparent. However, it is instructive to contemplate the hypothetical scenario of temporally reversing this diffusion process. What if one could initiate this process with a uniform distribution and thereby generate the original data distribution? Within the framework of conventional physics, the spontaneous reversal of such diffusion processes is exceedingly improbable—liquids do not spontaneously separate any more than shattered glass spontaneously reconstitutes its original structure. It is at this juncture that the paradigm of machine learning enters the discourse, offering a potential way to address

these challenges.

6.2 Forward diffusion process

Let's consider the data distribution $x_0 \sim q(x_0)$. In the forward diffusion process, noise is gradually added to a clean datum, until its structure is destroyed and the datum is transformed into pure noise. In other words, the forward process is characterised by a Markov chain, which generates a sequence of random variables $x_1, x_2, \dots x_T$ with transition kernel $q(x_t|x_{t-1})$. We define a variance schedule $\{\beta_t \in (0,1)\}_{t=0}^{T}$, such that the noise perturbing the data at each discrete timestep t is an isotropic Gaussian with variance β_t. The Markov chain is defined by the transition kernel:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}) \tag{6.1}$$

The joint distribution of $x_1, x_2, \dots x_T$ conditioned on x_0 is:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{6.2}$$

The transition kernel is often handcrafted to transform the initial data distribution into a tractable prior distribution. Thus, the forward diffusion process can be easily implemented. The diffusion model aims to learn the distribution of the reverse process $q(x_{t-1}|x_t)$, which is unknown.

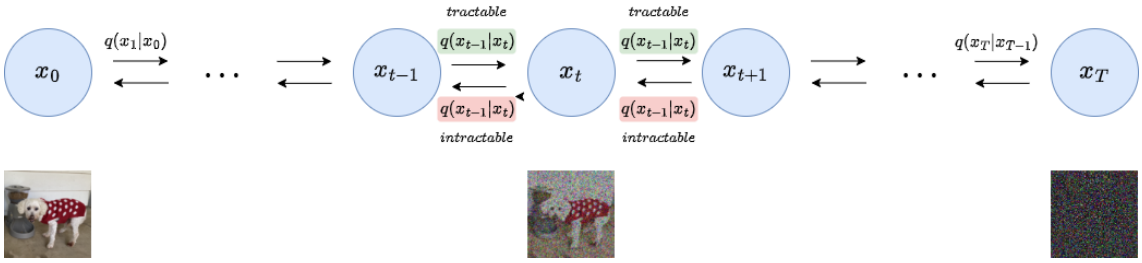

Figure 6.1. *Forward and Reverse Diffusion Processes [2].*

Lemma 6.1 (Reparameterisation Trick)**.** *Let x be a random variable drawn from a normal distribution with arbitrary mean μ and variance σ^2, i.e. $x \sim q(x|\mu) = \mathcal{N}(x; \mu, \sigma^2)$. Then, x can be equivalently expressed in the form:*

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

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

Proposition 6.1. *Let x_t be drawn from $q(x_t|x_{t-1})$ at timestep t. Then x_t can be directly expressed as a linear combination of x_0 and a noise variable ϵ:*

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

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

Proof. Let $t \in [0, 1, 2, \ldots, T]$. Then, by Eq. (6.1): $x_t \sim q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$. Now, let $a_t = 1-\beta_t$ and $\bar{a}_t = \prod_{i=1}^{t} a_i$.

According to Lemma 6.1, $x_t = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) = \mathcal{N}(x_t; \sqrt{a_t}x_{t-1}, (1-a_t)\mathbf{I})$ can be expressed as follows:

$$x_t = \sqrt{a_t}x_{t-1} + \sqrt{1-a_t}\epsilon_{t-1} \qquad \text{with } \epsilon_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

and similarly, $x_{t-1} \sim q(x_{t-1}|x_{t-2})$ can be rewritten as:

$$x_{t-1} = \sqrt{a_{t-1}}x_{t-2} + \sqrt{1-a_{t-1}}\epsilon_{t-2} \qquad \text{with } \epsilon_{t-2} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

As a result, x_t becomes:

$$\begin{aligned} x_t &= \sqrt{a_t}x_{t-1} + \sqrt{1-a_t}\epsilon_{t-1} \\ &= \sqrt{a_t}\left[\sqrt{a_{t-1}}x_{t-2} + \sqrt{1-a_{t-1}}\epsilon_{t-2}\right] + \sqrt{1-a_t}\epsilon_{t-1} \\ &= \sqrt{a_t a_{t-1}}x_{t-2} + \underbrace{\sqrt{a_t - a_t a_{t-1}}\epsilon_{t-2} + \sqrt{1-a_t}\epsilon_{t-1}} \end{aligned}$$

The underbraced quantity is the sum of two independent Gaussian random variables, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$. It remains a Gaussian with mean being the sum of the two means, and variance being the sum of the two variances, i.e. $\mathcal{N}(0, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$. $\sqrt{1-a_t}\epsilon_{t-1}$ is a sample from Gaussian $\mathcal{N}(0, (1-a_t)\mathbf{I})$, and $\sqrt{a_t - a_t a_{t-1}}\epsilon_{t-2}$ is a sample from Gaussian $\mathcal{N}(0, (a_t - a_t a_{t-1})\mathbf{I})$. As a result, we can treat their sum as a random variable sampled from Gaussian $\mathcal{N}(0, (1 - a_t + a_t - a_t a_{t-1})\mathbf{I}) = \mathcal{N}(0, (1 - a_t a_{t-1})\mathbf{I})$. Therefore, the last expression can be rewritten, using the reparameterisation trick as:

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

By repeating the above process, x_t takes this final form:

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

□

According to proposition 6.1, the distributions $\mathcal{N}(x_t; \sqrt{a_t}x_{t-1}, (1-a_t)\mathbf{I})$ and $\mathcal{N}(x_t; \sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)\mathbf{I})$ are equivalent. This is significant, as it enables sampling from any intermediate distribution of the forward diffusion process in a single step.

6.3 Reverse diffusion process

The reverse diffusion process can be utilised for the generation of new data samples. Diffusion models start by generating an initial Gaussian sample $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Noise is then gradually removed by running a learnable Markov chain in the reverse time direction. Sohl-Dickstein *et al.* show that if β_t is small enough, then $q(x_{t-1}|x_t)$ will also be a Gaussian distribution [6]. To perform reverse diffusion, one would have to model the kernels of the reverse timesteps. Estimating $q(x_{t-1}|x_t)$ is unfeasible, as it requires using the entire training dataset. A learnable transition kernel p_∂ is utilised instead, to approximate the conditional probabilities. The kernel takes the following form:

$$p_\partial(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\partial(x_t, t), \Sigma_\partial(x_t, t)) \tag{6.3}$$

∂ denotes model parameters and the mean $\mu_\partial(x_t, t)$ and variance $\Sigma_\partial(x_t, t)$ are parameterised by neural networks.

The joint distribution of the reverse Markov chain is:

$$p_\partial(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\partial(x_{t-1}|x_t) \tag{6.4}$$

The key to making this sampling process work effectively is to train the reverse Markov chain so that it mimics the time reversal of the forward Markov chain. In other words, the ∂ parameter must be adjusted to make sure that the reverse Markov chain's behaviour closely resembles that of the forward process. This adjustment should be such that the joint distribution of the reverse Markov chain $p_\partial(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\partial(x_{t-1}|x_t)$ (Eq. 6.4) closely approximates that of the forward process $q(x_{0:T}) = q(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1})$.

Thus far, $q(x_{t-1}|x_t)$ is not tractable, unless we condition it on x_0. We get:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathbf{I}) \tag{6.5}$$

By applying Bayes' rule, $q(x_{t-1}|x_t, x_0)$ becomes:

$$\begin{aligned}
q(x_{t-1}|x_t, x_0) &= \frac{q(x_t|x_{t-1}, x_0) q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&= \frac{\mathcal{N}(x_t; \sqrt{a_t} x_{t-1}, (1-a_t)\mathbf{I}) \mathcal{N}(x_{t-1}; \sqrt{\bar{a}_{t-1}} x_0, (1-\bar{a}_{t-1})\mathbf{I})}{\mathcal{N}(x_t; \sqrt{\bar{a}_t} x_0, (1-\bar{a}_t)\mathbf{I})} \\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{(x_t - \sqrt{a_t} x_{t-1})^2}{1-a_t} + \frac{(x_{t-1} - \sqrt{\bar{a}_{t-1}} x_0)^2}{1-\bar{a}_{t-1}} - \frac{(x_t - \sqrt{\bar{a}_t} x_0)^2}{1-\bar{a}_t}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\frac{(-2\sqrt{a_t} x_t x_{t-1} + a_t x_{t-1}^2)}{1-a_t} + \frac{(x_{t-1}^2 - 2\sqrt{\bar{a}_{t-1}} x_{t-1} x_0)}{1-\bar{a}_{t-1}} + C(x_t, x_0)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[-\frac{2\sqrt{a_t} x_t x_{t-1}}{1-a_t} + \frac{a_t x_{t-1}^2}{1-a_t} + \frac{x_{t-1}^2}{1-\bar{a}_{t-1}} - \frac{2\sqrt{\bar{a}_{t-1}} x_{t-1} x_0}{1-\bar{a}_{t-1}}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\left(\frac{a_t}{1-a_t} + \frac{1}{1-\bar{a}_{t-1}}\right) x_{t-1}^2 - 2\left(\frac{\sqrt{a_t} x_t}{1-a_t} + \frac{\sqrt{\bar{a}_{t-1}} x_0}{1-\bar{a}_{t-1}}\right) x_{t-1}\right]\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \exp\left\{-\frac{1}{2}\left[\frac{a_t(1-\bar{a}_{t-1})+1-a_t}{(1-a_t)(1-\bar{a}_{t-1})}x_{t-1}^2 - 2\left(\frac{\sqrt{a_t}x_t}{1-a_t}+\frac{\sqrt{\bar{a}_{t-1}}x_0}{1-\bar{a}_{t-1}}\right)x_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\frac{a_t-\bar{a}_t+1-a_t}{(1-a_t)(1-\bar{a}_{t-1})}x_{t-1}^2 - 2\left(\frac{\sqrt{a_t}x_t}{1-a_t}+\frac{\sqrt{\bar{a}_{t-1}}x_0}{1-\bar{a}_{t-1}}\right)x_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\frac{1-\bar{a}_t}{(1-a_t)(1-\bar{a}_{t-1})}x_{t-1}^2 - 2\left(\frac{\sqrt{a_t}x_t}{1-a_t}+\frac{\sqrt{\bar{a}_{t-1}}x_0}{1-\bar{a}_{t-1}}\right)x_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left(\frac{1-\bar{a}_t}{(1-a_t)(1-\bar{a}_{t-1})}\right)\left[x_{t-1}^2 - 2\frac{\left(\frac{\sqrt{a_t}x_t}{1-a_t}+\frac{\sqrt{\bar{a}_{t-1}}x_0}{1-\bar{a}_{t-1}}\right)}{\frac{1-\bar{a}_t}{(1-a_t)(1-\bar{a}_{t-1})}}x_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left(\frac{1-\bar{a}_t}{(1-a_t)(1-\bar{a}_{t-1})}\right)\left[x_{t-1}^2 - 2\frac{\left(\frac{\sqrt{a_t}x_t}{1-a_t}+\frac{\sqrt{\bar{a}_{t-1}}x_0}{1-\bar{a}_{t-1}}\right)(1-a_t)(1-\bar{a}_{t-1})}{1-\bar{a}_t}x_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left(\frac{1}{\frac{(1-a_t)(1-\bar{a}_{t-1})}{1-\bar{a}_t}}\right)\left[x_{t-1}^2 - 2\frac{\sqrt{a_t}(1-\bar{a}_{t-1})x_t+\sqrt{\bar{a}_{t-1}}(1-a_t)x_0}{1-\bar{a}_t}x_{t-1}\right]\right\} \\
&\propto \mathcal{N}(x_{t-1}; \underbrace{\frac{\sqrt{a_t}(1-\bar{a}_{t-1})x_t+\sqrt{\bar{a}_{t-1}}(1-a_t)x_0}{1-\bar{a}_t}}_{\tilde{\mu}_t(x_t,x_0)}, \underbrace{\frac{(1-a_t)(1-\bar{a}_{t-1})}{1-\bar{a}_t}}_{\tilde{\beta}_t}\mathbf{I})
\end{aligned}
$$

where $C(x_t, x_0)$ is a constant term with respect to x_{t-1}, as it involves only x_t, x_0 and a values. The term is implicitly returned in the final line to complete the square [2].

We can rearrange the equation in Proposition 6.1, to show that x_0 can be expressed in the following form:

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

By plugging this quantity into our previous derivation of $\tilde{\mu}_t(x_t, x_0)$, we get:

$$
\begin{aligned}
\tilde{\mu}_t(x_t, x_0) &= \frac{\sqrt{a_t}(1-\bar{a}_{t-1})}{1-\bar{a}_t}x_t + \frac{\sqrt{\bar{a}_{t-1}}(1-a_t)}{1-\bar{a}_t}\frac{1}{\sqrt{\bar{a}_t}}(x_t - \sqrt{1-\bar{a}_t}\epsilon_t) \\
&= \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right)
\end{aligned}
$$

Therefore, we can set our approximate denoising transition mean as:

$$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right) \tag{6.6}$$

and the transition variance as:

$$
\begin{aligned}
\tilde{\beta}_t &= \frac{(1-a_t)(1-\bar{a}_{t-1})}{1-\bar{a}_t} \\
&= \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\beta_t
\end{aligned} \tag{6.7}
$$

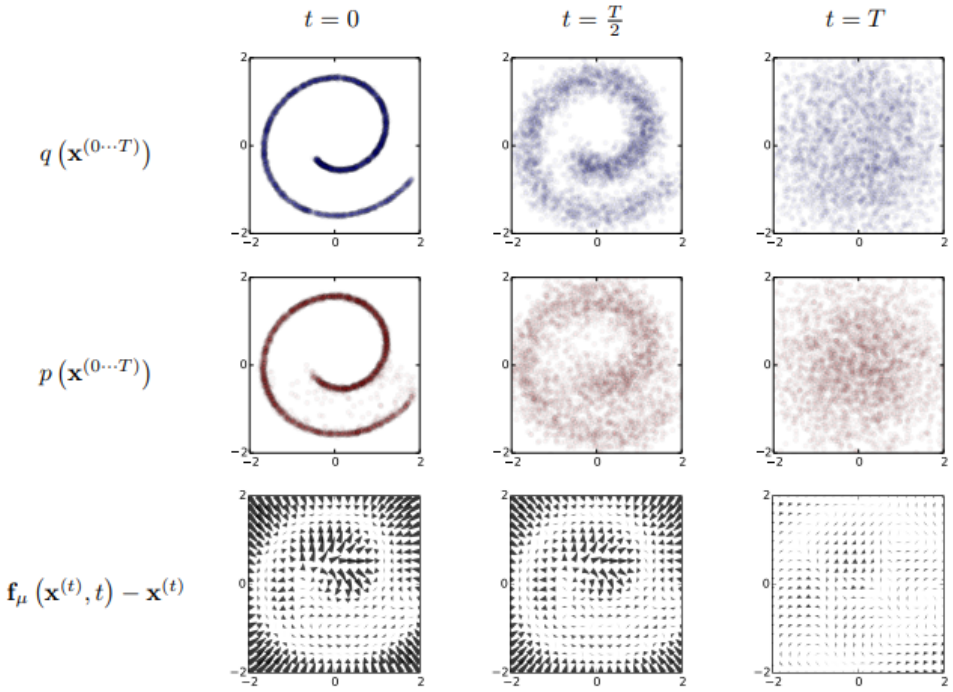

Figure 6.2. *Swiss roll example*
The proposed modeling framework applied to 2-dimensional Swiss roll data. The top row showcases time slices from the forward trajectory, highlighting the transition from the original data distribution to an identity-covariance Gaussian through Gaussian diffusion. The middle row illustrates the reverse trajectory, while the bottom row depicts the drift term for the reverse diffusion process. [6]

6.4 Training objective

To train the reverse Markov chain to match the time reversal of the forward Markov chain, we must adjust the ∂ parameter, so that the joint reverse distribution:

$$p_\partial(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\partial(x_{t-1}|x_t) \tag{6.8}$$

closely approximates the joint distribution of the forward process:

$$q(x_{0:T}) = q(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{6.9}$$

We employ our model to approximate $\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right)$. Since we have access to x_t at training time, we train our model to predict the Gaussian noise term from input x_t, at timestep t:

$$\mu_\partial(x_t, t) = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_\partial(x_t, t)\right) \tag{6.10}$$

x_{t-1} becomes:

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

Ho *et al.* [12] parameterise the loss term L_t to minimise the difference from $\tilde{\mu}_t$:

$$\begin{aligned}
L_t &= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta(x_t,t)\|_2^2}\|\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)\|^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta(x_t,t)\|_2^2}\|\frac{1}{\sqrt{a_t}}\left(x_t-\frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_t\right)-\frac{1}{\sqrt{a_t}}\left(x_t-\frac{1-a_t}{\sqrt{1-\bar{a}_t}}\epsilon_\theta(x_t,t)\right)\|^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{(1-a_t)^2}{2a_t(1-\bar{a}_t)\|\Sigma_\theta(x_t,t)\|_2^2}\|\epsilon_t-\epsilon_\theta(x_t,t)\|^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{(1-a_t)^2}{2a_t(1-\bar{a}_t)\|\Sigma_\theta(x_t,t)\|_2^2}\|\epsilon_t-\epsilon_\theta(\sqrt{\bar{a}_t}x_0+\sqrt{1-\bar{a}_t}\epsilon_t,t)\|^2\right]
\end{aligned}$$

6.4.1 Simplified training objective

Ho *et al.* [12] found that omitting the weighting term from the training objective is easier to implement and benefits sample quality:

$$\begin{aligned}
L_t^{simple} &= \mathbb{E}_{t,x_0,\epsilon_t}\left[\|\epsilon_t-\epsilon_\theta(x_t,t)\|^2\right] \\
&= \mathbb{E}_{t,x_0,\epsilon_t}\left[\|\epsilon_t-\epsilon_\theta(\sqrt{\bar{a}_t}x_0+\sqrt{1-\bar{a}_t}\epsilon_t,t)\|^2\right]
\end{aligned}$$

where t is uniform in $[1,T]$.

6.5 Denoising Diffusion Implicit Models

We have established that diffusion models produce samples using a Markov chain which starts from a pure noise sample and progressively denoises it into a new image. This Markov chain is obtained by approximately reversing the forward diffusion process, which consists of up to a few thousand steps, a task that requires many iterations. Due to this limitation, diffusion models were much slower, compared to other state-of-the-art generative models, such as Generative Adversarial Networks (GANs).

Song *et al.* introduced Denoising Diffusion Implicit Models (DDIMs) [1], which were the first implicit probabilistic models that could produce high quality samples. They improved sampling speed by extending the original DDPM to non-Markovian cases.

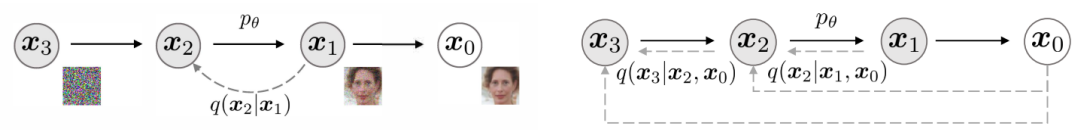

Figure 6.3. *Comparison of diffusion (left) and non-Markovian (right) inference models [1].*

6.5.1 Updated sampler

We now reparameterise the reverse diffusion distributions q_σ, using Proposition 6.1.

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

The degree of stochasticity in the forward process is determined by the magnitude of σ. As σ approaches 0, we reach an extreme scenario where, as long as we observe x_0 and x_t for some time t, the value of x_{t-1} becomes deterministically known and fixed.

Recalling Eq. 6.5 and Eq. 6.7, we have $q(x_{t-1}|x_t,x_0) = \mathcal{N}(x_{t-1};\tilde{\mu}_t(x_t,x_0),\tilde{\beta}_t\mathbf{I})$. Thus:

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

This setting corresponds to DDPMs

We set $\sigma_t^2 = \eta\cdot\tilde{\beta}_t$, $\eta\in\mathbb{R}^+$. This formulation allows us to control the stohasticity of the procedure, using the η hyperparameter. Denoising DIffusion Implicit Models (DDIMs) correspond to setting $\eta = 0$. In this case, sampling becomes deterministic and samples are generated from latent variables with a fixed procedure (from x_T to x_0). The DDIM name derives from the fact that the model is is an implicit probabilistic model trained with the DDPM objective, even though the reverse process is no longer a diffusion.

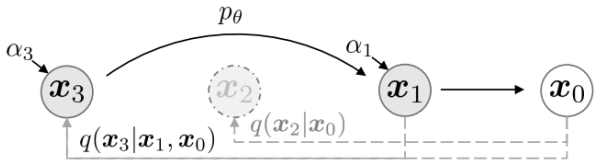

Figure 6.4. *Accelerated generation using DDIM [1].*

It has been empirically observed [1] that executing DDIM on a subset $S = \{\tau_1,\dots,\tau_S\}$ of the initial T training steps results in optimal performance. Thus, the inference process becomes:

$$
q_{\sigma,\tau}(x_{\tau_{i-1}}|x_{\tau_i},x_0) = \mathcal{N}\left(x_{\tau_{i-1}};\ \sqrt{\bar{a}_{\tau_{i-1}}}x_0 + \sqrt{1-\bar{a}_{\tau_{i-1}}-\sigma_{\tau_i}^2}\cdot\frac{x_{\tau_i}-\sqrt{\bar{a}_{\tau_i}}x_0}{\sqrt{1-\bar{a}_{\tau_i}}},\ \sigma_{\tau_i}^2\mathbf{I}\right)\quad \forall i\in[S]
$$

It is clear that the reason for DDIM's significant speed improvement is its capability to train the diffusion model on a large number of forward steps while generating samples from a selected subset of steps. On the other hand, DDPM performs best when applied on all initial T timesteps. Through DDIMs, an inherent trade-off emerges between sampling quality and computational costs. Namely, one can increase the number of steps taken during sampling, in order to increase sample quality, or decrease the number of steps to achieve faster inference. This trade-off is evident in Fig. 6.5 and Fig. 6.6.

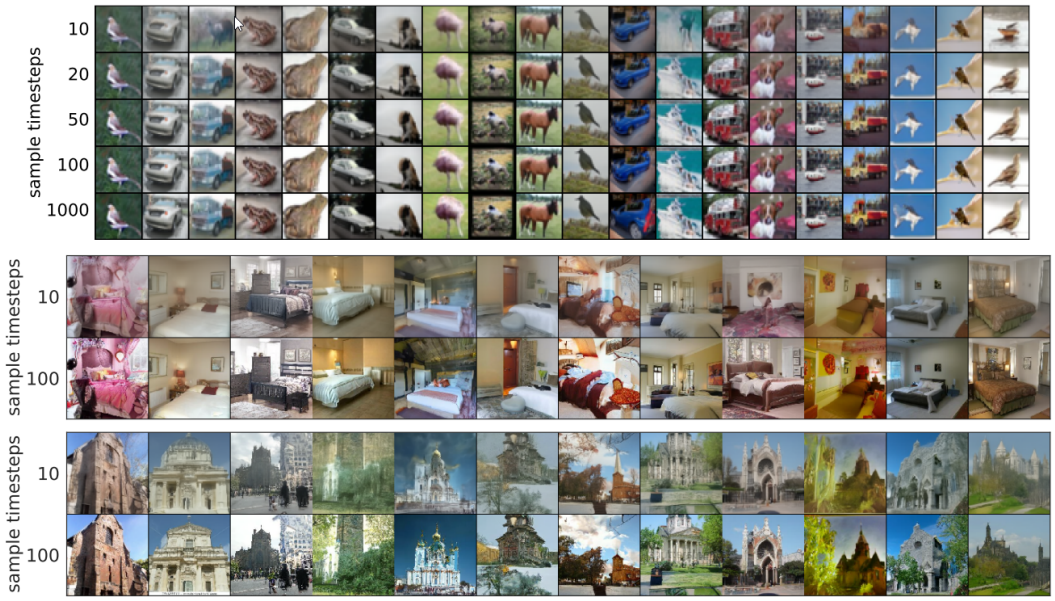

Figure 6.5. *DDIM samples with the same random x_T and different number of steps [1].*

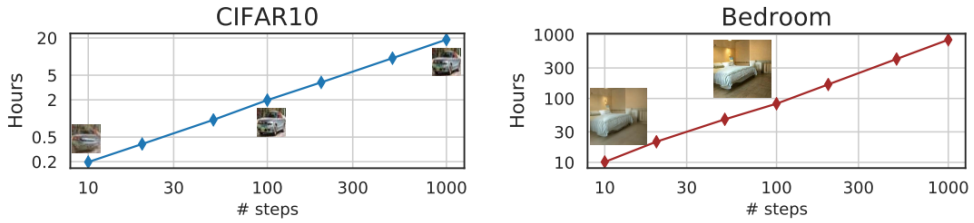

Figure 6.6. *Hours to sample 50k images using DDIM [1].*

6.6 Related Algorithms

Algorithms 6.1 and 6.2 are utilised to train and sample from a Denoising Diffusion Probabilistic Model. Algorithm 6.2 uses the simplified training objective presented in subsection 6.4.1.

ALGORITHM 6.1. *DDPM Training [12]*

repeat
  $x_0 \sim q(x_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\vartheta \left\| \epsilon - \epsilon_\vartheta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \right\|^2$
until converged

ALGORITHM 6.2. *DDPM Sampling [12].*

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $z = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\vartheta(x_t, t) \right) + \sigma_t z$
end for
return x_0

Chapter 7

Score-Based Generative Models

Another important member in the family of diffusion models is score-based generative models (SGMs) [4]. Given the data distribution $p_{data}(x)$, we define the score function as the gradient of the log probability density function, also known as the (Stein) score function. The key principle of SGMs is to estimate the score function, using a deep neural network, and generate samples using score-based sampling approaches.

Score-based models have achieved state-of-the-art results on tasks such as image [4, 15, 14, 12], and audio [8, 7] generation.

7.1 Score function and score matching for score estimation

Suppose our dataset consists of i.i.d. samples $\{x_i \in \mathbb{R}^D\}_{i=1}^N$ from a data distribution $p_{data}(x)$, which is unknown. The score of a probability density $p(x)$ is defined as $\nabla_x \log p(x)$. It is essentially a vector field, pointing in the direction where the log data density grows the most. The score network $s_\partial : \mathbb{R}^D \to \mathbb{R}^D$ is a neural network parameterized by ∂, which will be trained to approximate the score of $p_{data}(x)$.

Score matching was initially utilised to learn non-normalised statistical models based on i.i.d. samples from a data distribution. It can be repurposed in training a score network $s_\partial(x)$ to estimate $\nabla_x \log p_{data}(x)$ without estimating $p_{data}(x)$ first. The objective is to minimise

$$\frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\|s_\partial(x) - \nabla_x \log p_{data}(x)\|_2^2\right] \tag{7.1}$$

which is equivalent to the following up to a constant:

$$\mathbb{E}_{p_{data}(x)}\left[\mathrm{tr}(\nabla_x s_\partial(x)) + \frac{1}{2}\|s_\partial(x)\|_2^2\right] \tag{7.2}$$

where $\nabla_x s_\partial(x)$ denotes the Jacobian of $s_\partial(x)$.

While the expectation over $p_{data}(x)$ can be quickly estimated using data samples, score matching is not scalable to deep networks and high dimensional data due to the computational cost of $\mathrm{tr}(\nabla_x s_\partial(x))$. To circumvent this problem, we present two popular methods for large scale score matching.

7.1.1 Denoising score matching

Denoising score matching [16] perturbs the data point x with a pre-specified noise distribution $q_\sigma(\tilde{x}|x)$ and employs score matching to estimate the score of the perturbed data distribution, $q_\sigma(\tilde{x}) \triangleq \int q_\sigma(\tilde{x}|x)p_{data}(x)dx$. The objective was proved equivalent to:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{data}(x)}\left[\|s_\partial(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2\right] \tag{7.3}$$

The optimal score network, $s_{\partial^*}(x)$, satisfies $s_{\partial^*}(x) = \nabla_x \log q_\sigma(x)$ almost surely, while $s_{\partial^*}(x) = \nabla_x \log q_\sigma(x) \approx \nabla_x \log p_{data}(x)$ is true only when noise is small enough, such that $q_\sigma(x) \approx p_{data}(x)$.

7.1.2 Sliced score matching

Sliced score matching [17] uses random projections to approximate $\text{tr}(\nabla_x s_\partial(x))$ in score matching. The objective becomes:

$$\mathbb{E}_{p_v}\mathbb{E}_{p_{data}}\left[v^T\nabla_x s_\partial(x)v + \frac{1}{2}\|s_\partial(x)\|_2^2\right] \tag{7.4}$$

where p_v is a simple distribution of random vectors, *e.g.*, the multivariate standard normal. The term $v^T\nabla_x s_\partial(x)v$ can be efficiently computed by forward mode auto-differentiation. In contrast to denoising score matching, which estimates the score function of perturbed data, sliced score matching estimates the score function of the original distribution. Due to the forward mode auto-differentiation, sliced score matching requires around four times more computations than denoising score matching.

7.2 Sampling with Langevin Dynamics

Langevin Dynamics is capable of generating samples from a probability density $p(x)$, using only the score function $\nabla_x \log p(x)$. This is why estimating the score function is so important in score-based modelling. With an initial data point $\tilde{x}_0 \sim \pi(x)$ (π being a prior distribution) and step size $\epsilon > 0$, Langevin Dynamics recursively computes the following:

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x \log p(\tilde{x}_{t-1}) + \sqrt{\epsilon}z_t \tag{7.5}$$

where $z_t \sim N(\mathbf{0}, \mathbf{I})$. When $\epsilon \to 0$ and $T \to \infty$, the distribution of $\tilde{x}_T$ equals $p(x)$. Under some regularity conditions [18], $\tilde{x}_T$ becomes an exact sample from $p(x)$. On the other hand, when $\epsilon > 0$ and $T < \infty$, a Metropolis-Hastings update is needed to correct the error of Eq. 7.5. In practice, we set ϵ to a small number and T to a large number and safely ignore this error.

The process of sampling from Eq. 7.5 relies on the the score function $\nabla_x \log p(x)$. Therefore, to obtain samples from $p_{data}(x)$, we can initially train our score network to approximate $s_\partial(x) \approx \nabla_x \log p_{data}(x)$. Subsequently, we can use this approximation in conjunction with Langevin dynamics to generate samples. This fundamental concept constitutes the core principle of score-based generative modeling.

7.3 Challenges of score-based generative models

Score-based generative models face two major challenges, that prevent the naive application of the ideas described in the previous sections.

7.3.1 The manifold hypothesis

Data in the real world, as well as in many datasets, tend to concentrate on on low dimensional manifolds in a high dimensional space, the ambient space. This is a problem for score-based modelling, as the score $\nabla_x \log p_{data}(x)$ is undefined, when x is confined to a low dimensional manifold. Additionally, the score matching objective in Eq. 7.2 will be inconsistent, because it depends on the support of the data distribution being the whole space [4].

Perturbing the original data distribution with a small Gaussian noise, such that the perturbed distribution will have full support over $\mathbb{R}^D$, is an easy way to circumvent this problem. This can be observed in Fig. 7.1, where the sliced score matching objective fluctuates irregularly, when trained on the original data distribution, but converges if the data has been perturbed with small Gaussian noise.

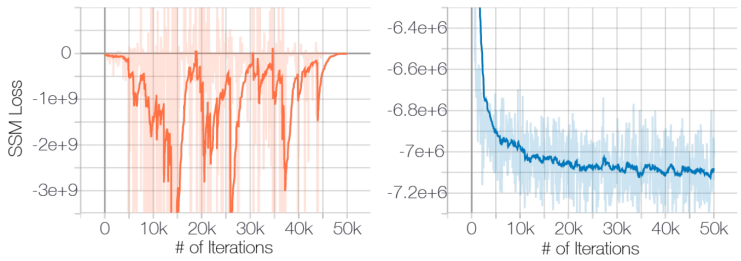

Figure 7.1. *Sliced score matching objective: No noise added (left); Data perturbed with* $\mathcal{N}(\boldsymbol{0}, 10^{-4})$ *(right) [4].*

7.3.2 Low data density regions

The lack of data samples in regions of low data density hinders the accurate estimation of the score function. This is a problem for two reasons. Firstly, the objective in score matching is to minimise $\frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\|s_\theta(x) - \nabla_x \log p_{data}(x)\|_2^2\right]$ (Eq. 7.1). The expectation over $p_{data}(x)$ is calculated using i.i.d. samples, thus score matching will not have sufficient data to estimate $\nabla_x \log p_{data}(x)$, in low data density regions. Secondly, when two modes of the data distribution are separated by low data density regions, Langevin Dynamics will not be able to recover their relative weights (since the gradient and log operators in the score function discard the weights), which could hinder convergence [4].

7.4 Noise Conditional Score Networks (NCSNs)

Song and Ermon [4] observed that perturbing the data with Gaussian noise can significantly aid score-based generative modelling in circumventing these challenges. The

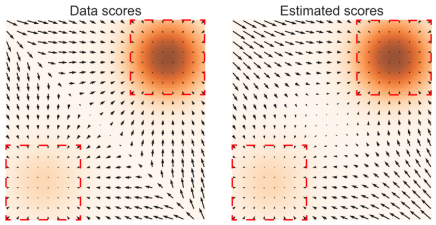

Figure 7.2. *Low data density example:* $\nabla_x \log p_{data}(x)$ *(left);* $s_\partial(x)$ *(right)*
Orange colourmap denotes the data density $p_{data}(x)$ *with darker colour implying higher data density. Within the red rectangles* $\nabla_x \log p_{data}(x) \approx s_\partial(x)$ *[4].*

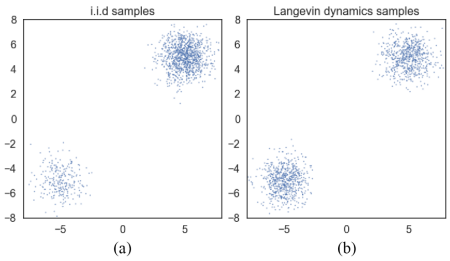

Figure 7.3. *Samples from a mixture of Gaussian with different methods: Exact sampling (left); Using Langevin dynamics (right)*
Langevin dynamics estimate the relative weights between the two modes incorrectly [4].

perturbed data will not be confined to a low dimensional manifold, since the support of the noise distribution is the entire space. Additionally, large Gaussian noise can fill the low data density regions in the original distribution and thus improve score estimation. Conversely, perturbing the data with large noise corrupts the distribution.

In order to attain the benefits of noise perturbation, without corrupting our original data distribution, we use a sequence of noise perturbed distributions, which converge to the true data distribution. This method, inspired by simulated annealing [19], benefits from the intermediate distributions and can improve the mixing rate of Langevin Dynamics.

Let $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$ be a sequence of L decreasing noise levels. We perturb the data with Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma_i \mathbf{I})$, using all $\{\sigma_i\}_{i=1}^L$, resulting in a sequence of perturbed distributions $q_{\sigma_i}(x) = \int p_{data}(t)\mathcal{N}(x; t, \sigma_i^2 \mathbf{I})dt$.

We need to estimate the score of each perturbed distribution. This can be achieved by training a conditional score network for all intermediate distributions: $\forall \sigma_i \in \{\sigma_i\}_{i=1}^L$: $s_\partial(x, \sigma_i) \approx \nabla_x \log q_{\sigma_i}(x)$. $s_\partial(x, \sigma)$ is called a *Noise Conditional Score Network (NCSN)* [4].

7.5 Training NCSNs via score matching

NCSNs can be trained by denoising, as well as sliced score matching. As Song and Ermon demonstrate [4], we opt for denoising score matching, as it is not only faster, but also naturally fit for the task of estimating the scores of noise perturbed data distributions.

We choose the noise distribution to be:

$$q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x, \sigma^2\mathbf{I}) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[\frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right] \tag{7.6}$$

Thus, the score will be:

$$\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = \nabla_x\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \frac{1}{2}\left(\frac{\tilde{x}-x}{\sigma}\right)^2\right] = -\frac{\tilde{x}-x}{\sigma^2} \tag{7.7}$$

The denoising score matching objective in Eq. 7.3 becomes:

$$\ell(\partial;\sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{data}(x)}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2\mathbf{I})}\left[\left\|s_\partial(\tilde{x},\sigma) + \frac{\tilde{x}-x}{\sigma^2}\right\|_2^2\right] \tag{7.8}$$

By combining Eq. 7.8 for all $\sigma \in \{\sigma_i\}_{i=1}^L$, we get the unified objective:

$$\mathcal{L}(\partial, \{\sigma_i\}_{i=1}^L) \triangleq \frac{1}{L}\sum_{i=1}^{L} \lambda(\sigma_i)\ell(\partial,\sigma_i) \tag{7.9}$$

where $\lambda(\sigma_i) > 0$ is a weighting coefficient, often chosen as $\lambda(\sigma_i) = \sigma_i^2$ [4].

7.6 NCSN inference via annealed Langevin Dynamics

Once our NCSN $s_\partial(x,\sigma)$ has been trained, we can use a modified version of Langevin Dynamics, called *annealed Langevin Dynamics*, for sampling. This method was inspired by simulated annealing [53] and annealed importance sampling [54]. The process begins by initialising samples from some fixed prior distribution. After each iteration, the method decreases the noise level σ_i and the step size a_i. The full algorithm is outlined below:

ALGORITHM 7.1. *Annealed Langevin Dynamics*

Require: $\{\sigma\}_{i=1}^L, \epsilon, T$
1: Initialise $\tilde{x}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad a_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ a_i is the step size.
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$ Draw $z_t \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
6: $\quad\quad \tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{a_i}{2}s_\partial(\tilde{x}_{t-1},\sigma_i) + \sqrt{a_i}z_t$
7: $\quad$ **end for**
8: $\quad \tilde{x}_0 \leftarrow \tilde{x}_T$
9: **end for**
return $\tilde{x}_T$

Fig. 7.4 shows three random sampling trajectories generated with Langevin dynamics, for a Mixture of Gaussians and starting from the same initialisation point. The left subplot shows these trajectories in 3D, while the right subplot shows them in 2D, against the ground-truth score function. Despite sharing a common intialisation point, the trajecto-

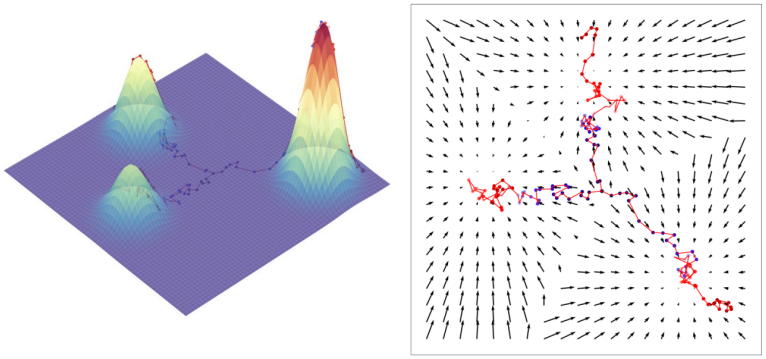

Figure 7.4. *Random sampling trajectories generated with Langevin dynamics [2].*

ries produce samples from different modes, thanks to the stochastic noise perturbations in Langevin dynamics. Otherwise, the procedure would always follow the score to the same mode deterministically.

Chapter 8

Stochastic Differential Equations

In the preceding chapters, we explored Denoising Diffusion Probabilistic Models and Score-Based Generative Models. Both classes of models rely on gradually perturbing data with increasing noise levels and learning to reverse this forward process to generate new samples from pure noise, that follow the original data distribution. Denoising Diffusion Probabilistic Models [6, 12] are probabilistic models, trained to reverse each step of the perturbation process, while Score-Based Generative Models [4] estimate the score function of the original data, perturbed at various noise scales, and utilise Annealed Langevin Dynamics to generate new data samples. Song *et al.* [3] demonstrated that both classes of models can be regarded as discretisations of stochastic differential equations, thus providing a unifying perspective on previous approaches.

This framework is comprised of an infinite number of distributions, which are used to diffuse the data into random noise. Thus, the forward diffusion process is given by a prescribed stochastic differential equation (SDE), which is independent of the data and has no trainable parameters. Given the score of the marginal probability densities as a function of time, and using the forward SDE, we can obtain the reverse-time SDE [20]. The reverse-time SDE can be approximated using a time-dependant neural network, to estimate the scores. Thus, we can generate new data samples, using numerical SDE solvers. These ideas are summarised in Fig. 8.1.

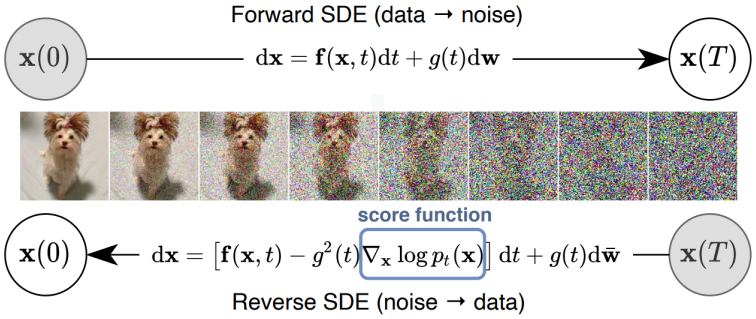

Figure 8.1. *Forward and Reverse SDE [3].*

8.1 Data perturbation using SDEs

Prior methods relied on perturbing data with various levels of noise. We expand on this concept by incorporating an infinite number of noise levels, resulting in perturbed data distributions that evolve according to an SDE, as the noise intensifies.

The diffusion process, which we examined in previous chapters, is the solution to the following SDE:

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

where w is the standard Wiener process (Brownian motion), $f(\cdot,t) : \mathbb{R}^D \to \mathbb{R}^D$ is a vector-valued function called the drift coefficient of $x(t)$ and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient of $x(t)$. The solution to the SDE, is a diffusion process $\{x(t)\}$, where $t \in [0,T]$, such that $x(0) \sim p_0$ (the data distribution) and $x(T) \sim p_T$ (the prior distribution).

We notice an analogy between the discrete steps of diffusion, discussed in previous chapters, and the continuous diffusion described by the SDE. Specifically, the discrete timesteps $i = 1,2,\dots,L$ are generalised to timesteps $t \in [0,T]$. Additionally, the discrete noise levels $\sigma_1, \sigma_2, \dots, \sigma_L$, which were hand picked in the discrete case, correspond to the SDE in the continuous case, which is also hand-designed. In other words, the way the SDE is designed is directly related to the noise levels we choose for perturbation in the discrete case. Similar to how in the discrete case, the noise levels are characteristic of the model, in the continuous case, the SDE is characteristic of the model.

In the following section, we explore the noise perturbations used in Score-Based Generative Models and Denoising Diffusion Probabilistic Models, which can be regarded as discretisations of *Variance Exploding (VE)* and *Variance Preserving (VP)* SDEs, respectively.

8.2 Variance Exploding and Variance Preserving SDEs

The noise perturbations used in Score-Based Generative Models and Denoising Diffusion Probabilistic Models are discretisations of two different types of SDEs [3].

In the case of Score-Based Generative Models, each perturbation kernel $p_{\sigma_i}(x|x_0)$ corresponds to the distribution of x_i, in the following Markov chain:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, z_{i-1}, \quad i = 1,\dots,N \tag{8.2}$$

where $z_{i-1} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Consider infinite noise scales ($N \to \infty$). In that case, $\{\sigma_i\}_{i=1}^N$ becomes a function $\sigma(t)$, the noise distribution z_i becomes $z(t)$ and the Markov chain $\{x_i\}_{i=1}^N$ becomes a continuous stochastic process. We use the notation $\{x(t)\}_{t=0}^1$, where t is a continuous time variable $t \in [0,1]$. The corresponding SDE, whose solution is $\{x(t)\}_{t=0}^1$, is:

$$dx = \sqrt{\frac{d\left[\sigma^2(t)\right]}{dt}}\, dw \tag{8.3}$$

For Denoising Diffusion Probabilistic Models, with perturbation kernels $\{p_{\alpha_i}(x|x_0)\}_{i=1}^N$,

the discrete Markov chain is:

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i}z_{i-1}, \quad i = 1, \dots, N \tag{8.4}$$

When $N \to \infty$, Eq. 8.4 converges to the following SDE:

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw \tag{8.5}$$

The noise perturbations used in Score-Based Generative Models are a discretisation of Eq. 8.3 and the noise perturbations used in Denoising Diffusion Probabilistic Models are a discretisation of Eq. 8.5.

Since the solution of Eq. 8.3 is always a process of exploding variance, as $t \to \infty$, the equation is known as the *Variance Exploding (VE)* SDE. Similarly, the solution of Eq. 8.5 is always a process of fixed variance of one, as long as the initial distribution has unit variance [3]. Thus, Eq. 8.5 is known as the *Variance Preserving (VP)* SDE.

8.3 Reversing the SDE to generate samples

We examined how, in the finite case, we can reverse the forward diffusion process to produce new samples, starting from prior distributions of pure noise. Likewise, in the continuous case, beginning with a sample of $x(T) \sim p_T$ and reversing the process of the SDE, we can generate new samples $x(0) \sim p_0$. Anderson [20] proved that reversing a diffusion process results in another diffusion process, which runs backwards in time and is given by the reverse-time SDE:

$$dx = \left[f(x,t) - g(t)^2\nabla_x \log p_t(x)\right]dt + g(t)d\bar{w} \tag{8.6}$$

Since this SDE runs backwards in time, dt represents an infinitesimal negative timestep, and $\bar{w}$ is the standard Wiener process, as time flows from T to 0. In order to sample from p_0, we must simulate the reverse SDE, which requires estimating the score function $\nabla_x \log p_t(x)$. Note that the score function matches the one learned by score-based models.

Once a score-based model has been trained to estimate the score function, we can utilise it to simulate the reverse SDE and solve it, to generate samples matching the original data distribution, i.e. p_0. We examine two approaches in solving the SDE, general purpose numerical SDE solvers and Predictor-Corrector (PC) samplers.

8.3.1 General purpose numerical SDE solvers

The reverse SDE can be solved using numerical SDE solvers, which are capable of approximating trajectories from SDEs. Any SDE solver can be utilised to produce samples. The most popular ones include the Euler-Maruyama and stochastic Runge-Kutta methods which correspond to discretisations of the stochastic dynamics.

8.3.2 Predictor-Corrector samplers

We can utilise the score-based model to improve the solutions to the reverse SDE. Predictor-Corrector samplers are comprised of two main components. The first is a predictor, which estimates the sample, while the second is a corrector, which corrects the marginal distribution of the predicted sample. These samplers are iterative, estimating both a prediction and a correction, at each timestep. The predictor is a numerical SDE solver, while the corrector is a score-based MCMC approach, such as Langevin MCMC.

Chapter 9

Discriminator Guidance

In an effort loosely inspired by Generative Adversarial Networks, Kim *et al.* [5] modify the diffusion sampling process by utilising a discriminator network to bridge the gap between the model score and the true data score. In their method, *Discriminator Guidance*, they maintain the pre-trained score model as a fixed component and introduce a discriminator network to classify real and generated data across different noise scales. The method incorporates a correction term into the model score, guided by the discriminator's feedback, which helps steer the sample generation towards more realistic paths.

9.1 Correction of Model Scores

Following the training of scores, we generate samples using the time-reversal generative process described by the reverse time SDE:

$$dx = \left[f(x,t) - g(t)^2 s_{\partial_\infty}(x,t)\right] dt + g(t)d\bar{w} \tag{9.1}$$

where s_{∂_∞} denotes the pre-trained score network. If the local optimum ∂_∞ deviates from the global optimum ∂_*, the generative process may differ from the reverse-time data process. Theorem 9.1 demonstrates that the generative process of Eq. 9.1 can align with the data process of Eq. 8.6 by adjusting the model score. The difference between the two processes is bridged by the *correction term*, which is nonzero whenever $\partial_\infty \neq \partial_*$.

Theorem 9.1. *Suppose p_{∂_∞} is the solution of the time-reversal generative process of Eq. 9.1. Let p_r^t and $p_{\partial_\infty}^t$ be the marginal densities (at t) of the forward-time SDE $dx = f(x,t)dt + g(t)dw$ starting from p_r and p_{∂_∞}, respectively. If $s_{\partial_\infty}(x,T) = \nabla \log \pi(x)$, where π is the prior distribution, and the log-likelihood $\log p_{\partial_\infty}$ equals its evidence lower bound $\mathcal{L}_{\partial_\infty}$, then the reverse-time SDE*

$$dx = \left[f(x,t) - g(t)^2 \nabla_x \log p_r^t(x)\right] dt + g(t)d\bar{w}$$

coincides with a diffusion process with an adjusted score,

$$dx = \left[f(x,t) - g(t)^2 (s_{\partial_\infty} + c_{\partial_\infty})(x,t)\right] dt + g(t)d\bar{w},$$

for $c_{\partial_\infty}(x,t) := \nabla \log \frac{p_r^t(x)}{p_{\partial_\infty}^t(x)}$.

9.2 Discriminator Guidance

The correction term $c_{\partial_\infty}(x,t) = \nabla \log \frac{p_r^t(x)}{p_{\partial_\infty}^t(x)}$ is generally intractable due to the inaccessible density ratio $\frac{p_r^t}{p_{\partial_\infty}^t}$. To address this, Kim *et al.* [5] approximate the density ratio by training a discriminator at every noise level t. For discriminator training, fake samples are generated from the process outlined in Eq. 9.1, equal in number to the actual data instances. The real and fake data are then classified using Binary Cross Entropy (BCE):

$$\begin{aligned}\mathcal{L}_\phi = \int \lambda(t)(&\mathbb{E}_{p_r^t(x)}[-\log d_\phi(x,t)] \\ &+ \mathbb{E}_{p_{\partial_\infty}^t(x)}[-\log(1-d_\phi(x,t))])dt,\end{aligned} \tag{9.2}$$

where λ represents the temporal weight.

Expressed in terms of the optimal discriminator ϕ_* derived from $\mathcal{L}_\phi$, the correction term is given by:

$$c_{\partial_\infty}(x,t) = \nabla \log \frac{d_{\phi_*}(x,t)}{1-d_{\phi_*}(x,t)}.$$

To estimate the correction term c_{∂_∞}, we utilise a neural discriminator ϕ:

$$c_{\partial_\infty}(x,t) \approx c_\phi(x,t) = \nabla \log \frac{d_\phi(x,t)}{1-d_\phi(x,t)}.$$

With this tractable estimate of the correction term, Discriminator Guidance (DG) is defined by:

$$dx = [f(x,t) - g^2(t)(s_{\partial_\infty} + c_\phi)(x,t)]dt + g(t)d\bar{w}. \tag{9.3}$$

9.2.1 Theoretical Analysis

While Discriminator Guidance was initially introduced in the context of stochastic differential equations, this section delves into the approach from the standpoint of statistical divergence between the data and sample distributions. Specifically, let $p_{\partial_\infty,\phi}$ be the sample distribution guided by the discriminator, as defined in Eq. 9.3. The central question becomes whether $p_{\partial_\infty,\phi}$ is closer to the data distribution p_r than p_{∂_∞}. This question is addressed in Theorem 9.2.

Theorem 9.2. *If the assumptions of Theorem 9.1 hold, then*

$$\begin{aligned}D_{KL}(p_r \| p_{\partial_\infty}) &= D_{KL}(p_r^T \| \pi) + E_{\partial_\infty}, \\ D_{KL}(p_r \| p_{\partial_\infty,\phi}) &\le D_{KL}(p_r^T \| \pi) + E_{\partial_\infty,\phi},\end{aligned}$$

where E_{∂_∞} is the score error

$$E_{\partial_\infty} = \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{p_r^t}[\|\nabla \log p_r^t - \mathbf{s}_{\partial_\infty}\|_2^2]dt,$$

and $E_{\partial_\infty,\phi}$ is the discriminator-adjusted score error

$$E_{\partial_\infty,\phi} = \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{p_r^t}[\|\nabla \log p_r^t - (\mathbf{s}_{\partial_\infty} + \mathbf{c}_\phi)\|_2^2]dt$$
$$= \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{p_r^t}[\|\mathbf{c}_{\partial_\infty} - \mathbf{c}_\phi\|_2^2]dt.$$

Table 9.1. *Discriminator-adjusted score error $E_{\partial_\infty,\phi}$ and corresponding Gain.*

Discriminator	$E_{\partial_\infty,\phi}$	Gain
Blind $d_{\phi_b}(\equiv 0.5)$	E_{∂_∞}	0
Optimal d_{ϕ_*}	0	E_{∂_∞} (Maximum)
Untrained $d_{\phi_0}(\approx 0.5)$	$\approx E_{\partial_\infty}$	≈ 0
Trained d_{ϕ_∞}	$\ll E_{\partial_\infty}$	$\nearrow E_{\partial_\infty}$

To assess the impact of discriminator training, we utilise Theorem 9.2 to compute the gain by subtracting two KL divergences,

$$D_{KL}(p_r\|p_{\partial_\infty,\phi}) \leq D_{KL}(p_r\|p_{\partial_\infty}) - \text{Gain}(\partial_\infty, \phi),$$

where $\text{Gain}(\partial_\infty, \phi) = E_{\partial_\infty} - E_{\partial_\infty,\phi}$ represents the difference between the score error and the discriminator-adjusted score error. While Theorem 9.2 doesn't ensure a strictly positive gain, Kim *et al.* demonstrate in their work [5] that the initialisation of the gain near zero gradually increases during discriminator training, as outlined in Table 9.1. Specifically, when the discriminator is untrained ($d_{\phi_0} \equiv 0.5$), no signal is derived from the discriminator gradient, and the discriminator-adjusted score error $E_{\partial_\infty,\phi_b}$ is equal to the score error E_{∂_∞}. Therefore, the gain is approximately zero when the discriminator is untrained ($d_{\phi_0} \approx 0.5$), as depicted in Fig. 9.1. Conversely, at the optimal discriminator d_ϕ, the neural correction $\mathbf{c}_{\phi_*}$ aligns with the target correction $\mathbf{c}_{\partial_\infty}$, satisfying $E_{\partial_\infty,\phi_*} = 0$ and allowing for the maximization of Gain as discriminator parameters are updated. Refer to Fig. 9.2 for a schematic representation.

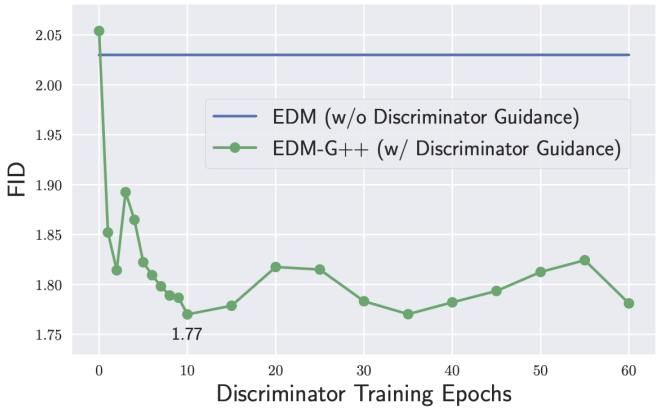

Figure 9.1. *Discriminator Guidance training and FID on CIFAR-10. [5].*

Essentially, Discriminator Guidance introduces an additional axial degree of freedom ϕ, which reparameterises the score error E_{∂_∞} into a discriminator-adjusted score error $E_{\partial_\infty,\phi}$. Consequently, the score error E_{∂_∞} is no longer optimised with the denoising score

loss $\mathcal{L}_\partial$, but the reparameterised error $E_{\partial_\infty,\phi}$ can be further optimised with an alternative loss $\mathcal{L}_\phi$ as defined in Eq. 9.2.

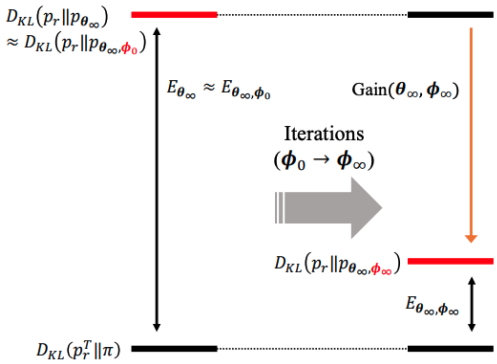

Figure 9.2. *Schematic illustration of Gain, which increases as discriminator is trained [5].*

Chapter 10

Exposure Bias

The iterative sampling chain in diffusion models is long, usually requiring thousands of steps due to the Gaussian assumption of reverse diffusion, which only holds for small step sizes [6]. This leads to the exposure bias problem, illustrated by Ning *et al.* [21]. Exposure bias refers to the discrepancy between the input data during training and inference phases. During training, the model is consistently exposed to the ground truth training sample x_t. However, during inference, the model relies on the previously generated sample, $\hat{x}_t$. This distinction between x_t and $\hat{x}_t$ results in a difference between $\epsilon_\partial(x_t)$ and $\epsilon_\partial(\hat{x}_t)$, where ϵ_∂ is the model's noise prediction. This disparity between the two predictions results in error accumulation and deviations in the sampling process, known as "sampling drift" [22]. Ning *et al.* [23] propose an effective method, *Epsilon Scaling*, for alleviating exposure bias, which is incorporated directly into the sampling process and requires no training or fine-tuning of the model.

10.1 Prediction Error leads to Exposure Bias

Ning *et al.* [21] identify a phenomenon associated with the sampling chain in diffusion models, which involves the accumulation of errors across T inference sampling steps. This accumulation is primarily attributed to the discrepancy between the training and inference stages. During training, the diffusion model is trained with a ground truth pair (x_t, x_{t-1}), learning to reconstruct x_{t-1} given x_t. However, during inference, the model lacks access to the ground truth x_t and relies on the previously generated $\hat{x}_t$, leading to a potential accumulation of errors. This mismatch between the input used in training and the input used in sampling resembles the exposure bias problem, originally observed in other generative models [24, 25].

During training, the ground truth training sample x_t is available to the model, with the training distribution being $q(x_t|x_0)$. In the inference phase, the model can only rely on the previously generated sample, $\hat{x}_t$. The sampling distribution can be denoted as $q(\hat{x}_t|x_{t+1}, x_\partial^{t+1})$, where x_∂^{t+1} is the prediction the model makes for x_0 given x_{t+1}, using Prop. 6.1, following notation by Ning *et al.* [23]. This results in a discrepancy between $\epsilon_\partial(x_t)$ and $\epsilon_\partial(\hat{x}_t)$.

Xiao *et al.* [26] observed that the sampling distribution $p_\theta(x_{t-1}|x_t)$ is parameterized as:

$$p_\theta(x_{t-1}|x_t) = q(x_{t-1}|x_t, x_\theta^t) \tag{10.1}$$

where x_θ^t represents the predicted x_0. The sampling process involves predicting ϵ using $\epsilon_\theta(x_t, t)$ and deriving the estimation x_θ^t for x_0 using Prop. 6.1. Then, x_{t-1} is generated based on the ground truth posterior $q(x_{t-1}|x_t, x_0)$, by replacing x_0 with x_θ^t. However, $q(x_{t-1}|x_t, x_0) = q(x_{t-1}|x_t, x_\theta^t)$ holds only if $x_\theta^t = x_0$. In practice, $q(x_{t-1}|x_t, x_0) \neq q(x_{t-1}|x_t, x_\theta^t)$, as the network makes prediction errors when estimating x_0, and, as a result, $q(x_{t-1}|x_t, x_\theta^t)$ does not have the same variance as $q(x_{t-1}|x_t, x_0)$.

Ning *et al.* [23] analytically calculate the discrepancy between the training and sampling distribution in DDPMs. They model x_θ^t as $p_\theta(x_0|x_t)$ and approximate it by a Gaussian distribution, following Bao *et al.* [55, 56].

$$x_\theta^t = x_0 + e_t \epsilon_0 \tag{10.2}$$

where e_t is the standard deviation of x_θ^t, and $\epsilon_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Their findings are summarized in Table 10.1. It becomes clear that the sampling distribution's variance is always larger than that of the training distribution by a factor of $(\frac{\sqrt{\bar{a}_t}\beta_{t+1}}{1-\bar{a}_{t+1}} e_{t+1})^2$. It must be noted that this is the prediction error produced in a single reverse diffusion step. During sampling, the errors accumulate across steps, resulting in the Exposure Bias problem.

	Mean	Variance
$q(x_t\|x_0)$	$\sqrt{\bar{a}_t}x_0$	$(1-\bar{a}_t)I$
$q(\hat{x}_t\|x_{t+1}, x_\theta^{t+1})$	$\sqrt{\bar{a}_t}x_0$	$(1-\bar{a}_t + (\frac{\sqrt{\bar{a}_t}\beta_{t+1}}{1-\bar{a}_{t+1}} e_{t+1})^2)I$

Table 10.1. *Mean and variance of $q(x_t|x_0)$ and $q(\hat{x}_t|x_{t+1}, x_\theta^{t+1})$*

10.2 Related Work

To address the exposure bias issue, Ning *et al.* [21] suggest explicitly modelling the prediction error during training. During the training phase, they perturb x_t as normal and provide the network with a new, noisier version of x_t. This simulates the training-sampling discrepancy, fooling the learned network into considering potential prediction errors during inference. Even though their method proves effective in reducing the exposure bias phenomenon, it is cumbersome as it necessitates retraining the score network entirely, a computationally expensive endeavour.

Li *et al.* [22] propose a different approach, which involves shifting the timestep t during sampling. They observe that the time step t is directly linked to the corruption level of data samples and demonstrate that adjusting the subsequent time step $t-1$ during sampling, based on the variance of the currently generated samples, can effectively mitigate exposure bias. Despite the fact that their method circumvents the need for model retraining, tuning the timestep shift is difficult to optimise.

10.3 Epsilon Scaling

Ning *et al.* [23] propose scaling down the predicted noise factor ϵ_θ^s (where s denotes the noise factor predicted in the sampling stage) by a factor λ_t at time step t as a way to reduce Exposure Bias. Their approach is based on the assumption that the accuracy of the prediction ϵ_θ^s can be enhanced if we are able to shift the input $(\hat{x}_t, t)$ away from the unreliable vector field (depicted as the orange curve in Fig. 11.1 and 11.2) and towards the dependable vector field (represented by the green curve in Fig. 11.1 and 11.2).

Their approach is rooted in the following observation: ϵ_θ^s and ϵ_θ^t (where t denotes to the noise factor predicted in the training stage) both originate from the same input $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at time step $t = T$. However, starting from time step $T-1$, $\hat{x}_t$ (the input for ϵ_θ^s) begins to deviate from x_t (the input for ϵ_θ^t) due to the $\epsilon_\theta(\cdot)$ error made in the previous time step. This iterative process continues throughout the sampling chain, leading to exposure bias. Therefore, we can bring $\hat{x}_t$ closer to x_t by reducing the overestimated magnitude of ϵ_θ^s. Their sampling method only differs from Eq. 6.10 in the λ_t term and can be expressed as:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\frac{\epsilon_\theta(x_t, t)}{\lambda_t}\right) \tag{10.3}$$

As a result, epsilon scaling is a plug-in method which requires no retraining or fine-tuning of the original score model and adds no overhead computational cost.

Chapter 11

Our Method

11.1 Quantifying Exposure Bias

As Table 10.1 illustrates, the x_t seen by the network during training differs from the $\hat{x}_t$ seen by the network during sampling. This discrepancy leads to a drift between the noise prediction made during training, ϵ^t_∂, and the noise prediction made during sampling, ϵ^s_∂. Following Ning *et al.* [23], we choose to measure the sampling drift at each timestep as the difference between ϵ^t_∂ and ϵ^s_∂. However, since the ground truth of ϵ^s_∂ is intractable in the sampling phase, we use the $L2$-norm to quantify the exposure bias [23].

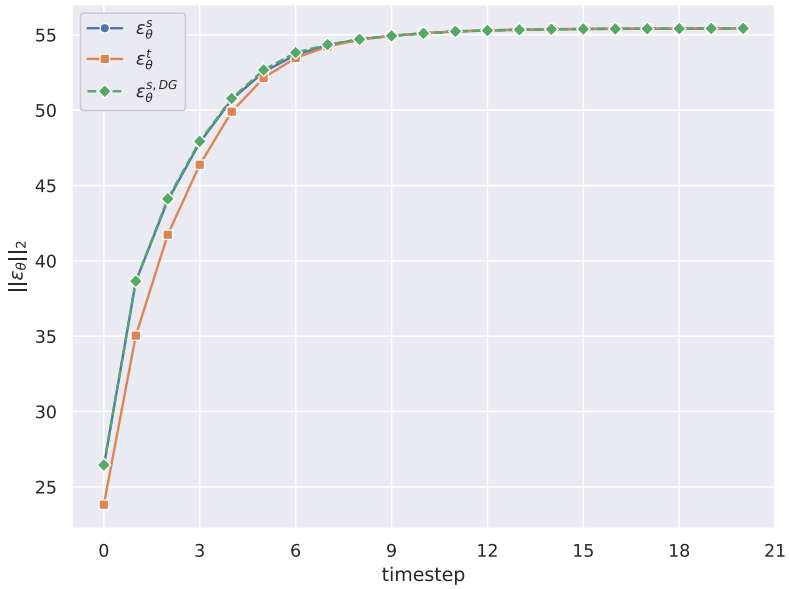

Figure 11.1. *EDM model, Euler 1st order solver. L2-norm of $\epsilon_\partial(\cdot)$ during 21-step sampling, sampling with DG and training. Statistical L2-norm was calculated using 50k samples at each timestep. Sampling is from right to left.*

In Fig. 11.1 and Fig. 11.2 we plot the $L2$-norm of ϵ^t_∂, ϵ^s_∂ and $\epsilon^{s,DG}_\partial$ using the Euler, as well as the Heun ODE solver. ϵ^s_∂ and $\epsilon^{s,DG}_\partial$ refer to the noise prediction in the vanilla EDM model and in the discriminator guided EDM-G++ model, respectively. We observe that in the case of the Euler solver, the $L2$-norm of $\epsilon^{s,DG}_\partial$ is larger than that of ϵ^t_∂ and nearly

coincides with the $L2$-norm of ϵ_∂^s. In the case of the Heun 2nd order solver, the difference between the two norms is smaller, however, the $L2$-norm of $\epsilon_\partial^{s,DG}$ is, once again, larger than that of ϵ_∂^t and closer to that of ϵ_∂^s. This means that the correction term offered by discriminator guidance does not alleviate the sampling procedure of its collected exposure bias. Instead, the prediction errors accumulate and the learnt vector field $\epsilon_\partial^{s,DG}$ deviates from the desired sampling trajectory.

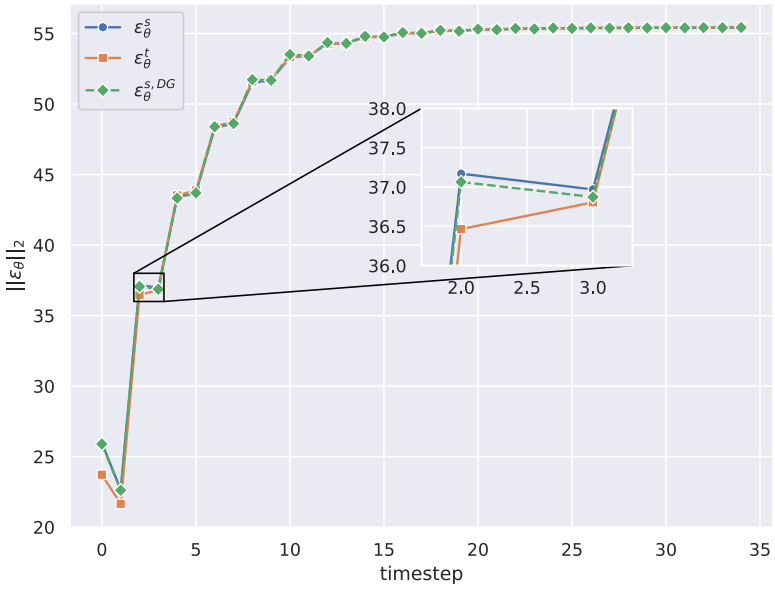

Figure 11.2. *EDM model, Heun 2nd order solver. L2-norm of $\epsilon_\partial(\cdot)$ during 35-step sampling, sampling with DG and training. Statistical L2-norm was calculated using 50k samples at each timestep. Sampling is from right to left.*

11.2 Proposed Framework

We use the EDM model, proposed by Karras *et al.* [14], as a score estimator due to the detailed way in which it was designed. Through careful network design and fine-tuning of hyperparameters, EDM achieves a substantial quality enhancement, reducing the FID score on the CIFAR-10 dataset to 1.97, demonstrating notable progress at the time. Moreover, seeing as Exposure Bias exhibits a strong correlation with FID score [23], we assume that in comparison to other networks, the vanilla EDM model demonstrates a reduced accumulation of exposure bias.

When it comes to the discriminator network, we follow the setup demonstrated by Kim *et al.* [5]. The discriminator network is comprised of the encoders of two U-Net structures. The first U-Net encoder is frozen and the second one is fine-tuned, to accelerate training. We leave the exploration of other discriminator architectures, such as Vision Transformers, as future work.

In Sec. 10.2, we mention different approaches which seek to reduce the impact of Exposure Bias on the sampling trajectory. Ning *et al.* [21] suggest introducing an extra

noise factor at each step during the training to mitigate the discrepancy between training and inference. However, this method proves cumbersome as it requires retraining the model from scratch. Li *et al.* [22] explore how manipulation of the time step during the reverse generation process can trick the model into reducing the Exposure Bias issue. This method is also difficult to implement as exploring the entire space of possible combinations is inconvenient as it requires costly experimentation to produce noteworthy results. The latest method to reduce Exposure Bias, introduced by Ning *et al.* [23], known as Epsilon Scaling, is our selected approach. It is a training free, plug-in method, which has proven effective in reducing Exposure Bias and significantly improving the FID score across a range of diffusion models (ADM [57], DDIM [1], EDM [14], LDM [33]). We present the main notion of Epsilon Scaling in Sec. 10.3.

Although the network output of EDM is the score function s_θ, not ϵ, the noise factor ϵ can easily be extracted at each sampling step and used to apply Epsilon Scaling [23].

When it comes to designing the scaling schedule λ_t, Ning *et al.* [23] propose that the term λ_t should be a linear function $\lambda_t = kt + b$ where k, b are constants. They also observe that the longer the sampling step, the smaller the k that should be used. Thus, they suggest a uniform schedule λ_t ($k = 0$) to facilitate practicality and simplify the exploration of the b parameter. In our experiments, we confirm that the use of a linear λ_t can provide similar or sub-optimal results, compared to the uniform schedule $\lambda_t = b$ and explore the constant scaling factor more extensively.

Chapter 12

Results

12.1 Euler Solver

We firstly present our results using the Euler 1st order ODE solver. When it comes to the Discriminator Guidance method (Kim *et al.* [5]), we identify a noteworthy omission. Namely, the authors do not report performance of the EDM-G++ model using the Euler ODE solver. We have calculated the FID score of EDM-G++ on the CIFAR-10 dataset using various numbers of timesteps and Discriminator Guidance weight, w^{DG} and we present our results in Fig. 12.1.

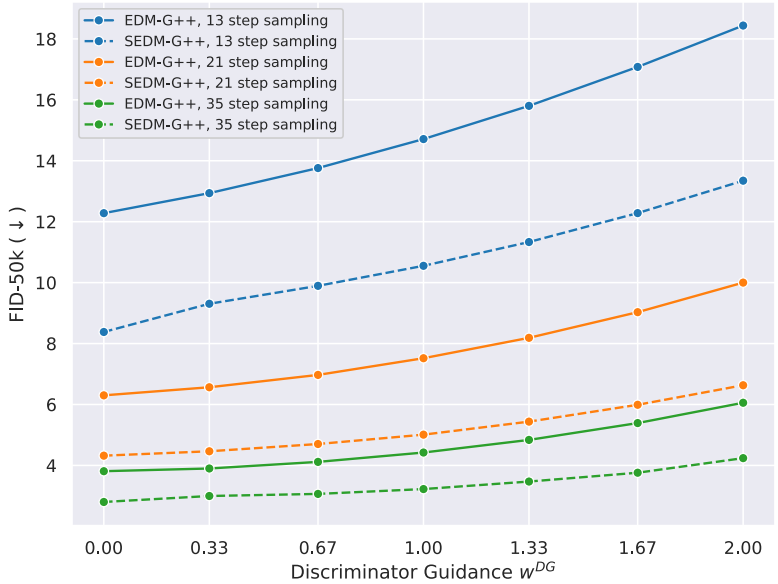

Figure 12.1. *FID-50k ablation study on DG weight (Euler Solver).*

Notably, the EDM-G++ model, which is guided by a discriminator, exhibits a noteworthy decrease in sample quality when compared to the baseline EDM model. This decline intensifies as the weight assigned to the discriminator's correction term, w^{DG}, increases. This outcome is intriguing in light of the fact that the inclusion of discriminator guidance led to a substantial reduction in the FID score using the Heun 2nd order ODE solver. It is plausible to surmise that the reduced FID score in the case of the 1st order solver may be

attributed to the absence of a corrective step in the sampling process. The enhancement of sample quality in the EDM model [14] is markedly facilitated by the inclusion of a corrective step in the ODE solver. This is a crucial factor contributing to the widespread adoption of the Heun ODE solver in recent research, as it consistently delivers superior performance [14].

Nevertheless, the epsilon scaling method demonstrates its efficacy in the context of discriminator-guided diffusion models as well. SEDM-G++ successfully reduces the FID score across different numbers of total timesteps and w^{DG} values when utilising the Euler ODE solver, as compared to the EDM-G++ and EDM baselines. Remarkably, SEDM-G++ narrows the performance gap between EDM-G++ using a 21-step Euler solver and a 35-step Euler solver, with the performance of SEDM-G++ using a 21-step Euler solver closely approaching that of the baseline EDM-G++ model using a 35-step Euler solver. This results in a significant reduction in computational and time requirements during inference, without compromising sample quality to a considerable extent.

In Figure 12.2, we present uncoordinated samples derived from the EDM-G++ baseline and our proposed SEDM-G++. Apart from the evident improvement in the FID score, our method yields noticeable qualitative enhancements in the generated samples. For example, the sample located in the third row and third column, as well as the sample in the fifth row and fourth column, exhibit a substantial enhancement compared to the baseline. In the left subfigure, the shapes appear blurry and obscure, making it challenging to discern the content of the images. Conversely, in the right subfigure, SEDM-G++ produces images with clear, well-defined shapes and vivid colors, rendering the identity of the depicted objects readily discernible.

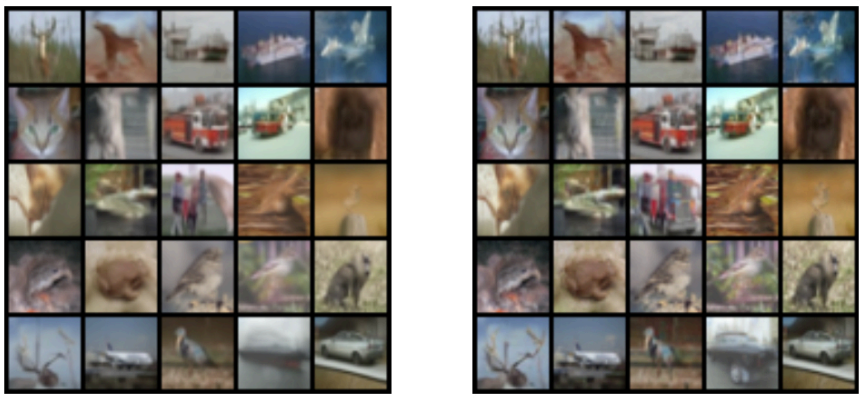

Figure 12.2. *Uncoordinated samples from the EDM-G++ baseline (left) and our proposed SEDM-G++ (right).*

12.2 Heun Solver

We further explore the performance of SEDM-G++ using the Heun 2nd order ODE solver. We conducted an ablation study on the relation between the weight attributed to discriminator guidance in the 1st order step of the Heun solver, namely $w^{DG}_{t,1st}$, and the epsilon scaling factor $\lambda_t = b$. Based on the work of Kim *et al.* [5], we set the 2nd order discriminator guidance weight, namely $w^{DG}_{t,2nd}$ equal to zero for all tests. This choice offers optimal sample quality and requires fewer computational resources, as the number of calls to the discriminator network is practically halved by omission in the 2nd order corrective steps. In order to reduce the computational needs of our study, rather than generating a total of 50k samples, we generate 10k samples for each setting and derive the FID-10k score, which suffices for the purpose of parameter optimisation [23]. The results are presented in Fig. 12.3.

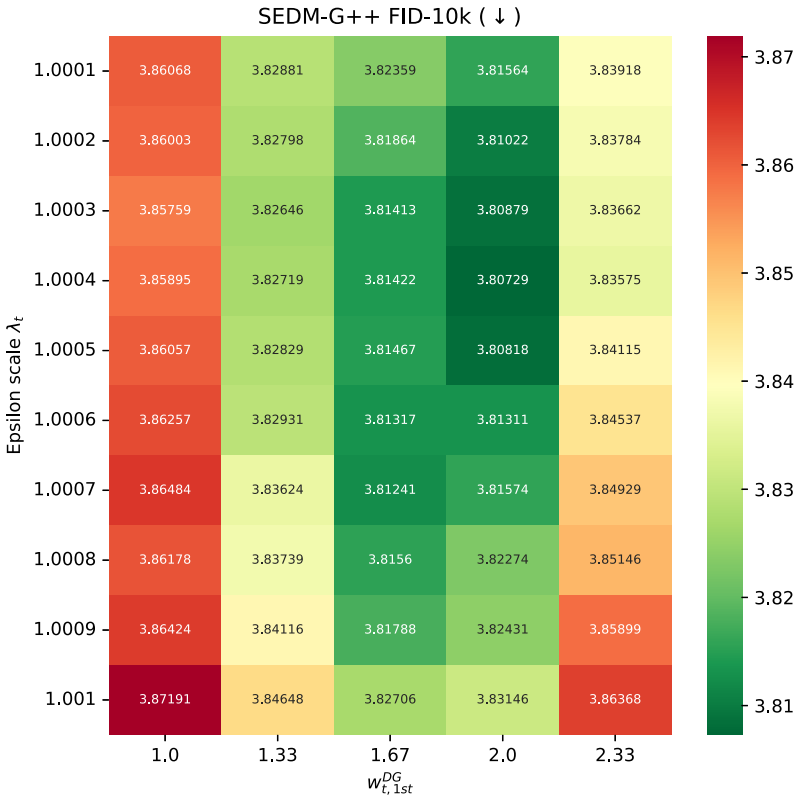

Figure 12.3. *FID-10k ablation study on DG weight and scaling factor (Heun Solver).*

Our observations indicate that the most effective discriminator guidance weight values are 1.67 and 2. Notably, when comparing these two values, we also note that the optimal epsilon scaling value λ_t decreases as the discriminator's weight coefficient increases. We further delve into the performance of the best-performing $w^{DG}_{t,1st}$ values through a comprehensive study, by generating 50k samples for each setting and utilising the proper FID-50k score to compare. We present our results in Fig. 12.4.

Our proposed SEDM-G++ outperforms the current state-of-the-art in unconditional CIFAR-10 image generation, by achieving an FID score of 1.73. The optimal hyperparameters used are $\hat{\eta}_t = 1.0004$, $w^{DG}_{t,1st} = 1.67$, and $w^{DG}_{t,2nd} = 0$. A comprehensive comparison between SEDM-G++ and other prominent diffusion models is provided in Table 12.1. Given that our approach is based on the EDM model [14], it maintains a low Number of Function Evaluations (NFE) at 35 network calls per batch. This figure is significant as it directly relates to the computational cost associated with the sampling process.

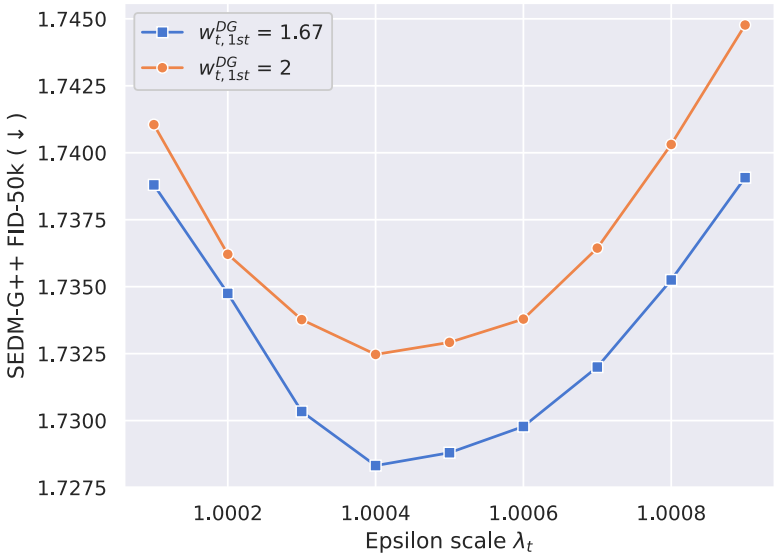

Figure 12.4. *FID-50k ablation study on best performing DG weight values (Heun Solver).*

Model	NFE(↓)	FID(↓)
VDM (Kingma *et al.*, 2021 [27])	1000	7.41
DDPM (Ho *et al.*, 2020 [12])	1000	3.17
iDDPM (Nichol & Dhariwal, 2021 [28])	1000	2.90
Soft Truncation (Kim *et al.*, 2022 [29])	2000	2.47
INDM (Kim *et al.*, 2022 [30])	2000	2.28
CLD-SGM (Dockhorn *et al.*, 2022 [31])	312	2.25
NCSN++ (Song *et al.*, 2020 [3])	2000	2.20
LSGM (Vahdat *et al.*, 2021 [32])	138	2.10
EDM (Karras *et al.*, 2022 [14])	**35**	1.97
EDM-G++ (Kim *et al.*, 2023 [5])	**35**	1.77
SEDM-G++ (ours)	**35**	**1.73**

Note: Following the work of Karras *et al.* [14], we calculate the FID for different seeds and report the minimum. Kim *et al.* [5] use random seeds for FID calculation. Manually calculating the FID of EDM-G++ results in an FID of 1.75.

Table 12.1. *FID-50k performance comparison on unconditional CIFAR-10 image generation.*

An intriguing observation pertains to the fact that the FID gain achieved through Epsilon Scaling in the Euler sampler is more pronounced compared to the case of the Heun sampler. This is in line with the observations of Ning *et al.* [23], who attribute this

phenomenon to two key factors. Firstly, higher-order ODE solvers, such as Heun solvers, entail a lower level of truncation error in contrast to Euler 1st order solvers. Secondly, the corrective steps integrated into the Heun solver serve to mitigate exposure bias by readjusting the drifted sampling trajectory back to the precise vector field. This is also evident in Fig. 11.1 and 11.2. In the case of the Heun solver, any prediction error (the root cause of exposure bias) incurred during each Euler step is rectified during the subsequent correction step (Fig. 11.2), leading to a reduction in exposure bias. This exposure bias perspective offers a comprehensive explanation for the superior performance of the Heun solver in diffusion models.

Chapter 13

Conclusion

Diffusion models have emerged as the dominant class of generative models, with applications in a wide range of tasks. Many recent works have shifted focus to refining sample quality through improvements in the sampling procedure using pre-trained score models. Inspired by Generative Adversarial Networks (GANs), Kim *et al.* [5] introduced the concept of *Discriminator Guidance* (DG), utilizing a discriminator network to bridge the gap between model score and true data score.

We explore the effectiveness of Discriminator Guidance in addressing exposure bias accumulation during the sampling process in diffusion models. Our findings reveal that, despite notable improvements in sample quality, Discriminator Guidance falls short in mitigating exposure bias. In response, we introduce SEDM-G++, a novel approach that integrates a modified sampling technique, incorporating both Discriminator Guidance and Epsilon Scaling [23]. Applying this method to the pre-trained EDM model, we demonstrate its consistent ability to enhance sample quality while reducing exposure bias. This improvement is observed across various ODE solvers, a range of numbers of timesteps employed, and different hyperparameter settings. Our proposed approach outperforms the current state-of-the-art, achieving an FID score of 1.73 on the unconditional CIFAR-10 dataset.

Appendices

Appendix A

Uncoordinated samples

Figure A.1. *Uncoordinated samples using EDM-G++, no Epsilon Scaling, Discriminator Guidance, Euler solver, 13 steps. FID: 17.08*

Figure A.2. *Uncoordinated samples using SEDM-G++, Epsilon Scaling, Discriminator Guidance, Euler solver, 13 steps. FID: 12.28*

Figure A.3. *Uncoordinated samples using EDM-G++, no Epsilon Scaling, Discriminator Guidance, Euler solver, 21 steps. FID: 9.03*

Figure A.4. *Uncoordinated samples using SEDM-G++, Epsilon Scaling, Discriminator Guidance, Euler solver, 21 steps. FID: 5.99*

Figure A.5. *Uncoordinated samples using EDM-G++, no Epsilon Scaling, Discriminator Guidance, Euler solver, 35 steps. FID: 5.39*

Figure A.6. *Uncoordinated samples using SEDM-G++, Epsilon Scaling, Discriminator Guidance, Euler solver, 35 steps. FID: 3.76*

Figure A.7. *Uncoordinated samples using EDM-G++, no Epsilon Scaling, Discriminator Guidance, Heun solver, 35 steps. FID: 1.77*

Figure A.8. *Uncoordinated samples using SEDM-G++, Epsilon Scaling, Discriminator Guidance, Heun solver, 35 steps. FID: 1.73*

Bibliography

[1] Jiaming Song, Chenlin Meng και Stefano Ermon. *Denoising Diffusion Implicit Models*. *International Conference on Learning Representations*, 2020.

[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*. *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[4] Yang Song και Stefano Ermon. *Generative modeling by estimating gradients of the data distribution*. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang και Il Chul Moon. *Refining Generative Process with Discriminator Guidance in Score-Based Diffusion Models*. *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, 2023.

[6] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan και Surya Ganguli. *Deep unsupervised learning using nonequilibrium thermodynamics*. *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, σελίδες 2256–2265, 2015.

[7] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao και Bryan Catanzaro. *DiffWave: A Versatile Diffusion Model for Audio Synthesis*. *International Conference on Learning Representations*, 2020.

[8] 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*, 2020.

[9] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi και David J Fleet. *Video Diffusion Models*. *Advances in Neural Information Processing Systems*, τόμος 35, σελίδες 8633–8646, 2022.

[10] Vikram Voleti, Alexia Jolicoeur-Martineau και Chris Pal. *MCVD-masked conditional video diffusion for prediction, generation, and interpolation*. *Advances in Neural Information Processing Systems*, 35:23371–23385, 2022.

[11] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler και Karsten Kreis. *Align your latents: High-resolution video synthesis*

with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 22563–22575, 2023.

[12] Jonathan Ho, Ajay Jain και Pieter Abbeel. *Denoising diffusion probabilistic models*. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, σελίδες 6840–6851, 2020.

[13] Alexander Quinn Nichol και Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. *Proceedings of the 38th International Conference on Machine Learning, ICML*, σελίδες 8162–8171, 2021.

[14] Tero Karras, Miika Aittala, Timo Aila και Samuli Laine. *Elucidating the design space of diffusion-based generative models*. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[15] Yang Song και Stefano Ermon. *Improved Techniques for Training Score-Based Generative Models*, 2020.

[16] Pascal Vincent. *A Connection Between Score Matching and Denoising Autoencoders*. *Neural Computation*, 23(7):1661–1674, 2011.

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

[18] Max Welling και Yee Teh. *Bayesian Learning via Stochastic Gradient Langevin Dynamics*. *Proceedings of the 28th International Conference on International Conference on Machine Learning*, σελίδες 681–688, 2011.

[19] Wikipedia. *Simulated Annealing*. Accessed on October 2, 2023.

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

[21] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara και Rita Cucchiara. *Input Perturbation Reduces Exposure Bias in Diffusion Models*. *International Conference on Machine Learning, ICML*, σελίδες 26245–26265, 2023.

[22] Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun και Marie Francine Moens. *Alleviating Exposure Bias in Diffusion Models through Sampling with Shifted Time Steps*, 2023.

[23] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah και Itir Onal Ertugrul. *Elucidating the Exposure Bias in Diffusion Models*, 2023.

[24] Florian Schmidt. *Generalization in Generation: A closer look at Exposure Bias*. *Proceedings of the 3rd Workshop on Neural Generation and Translation*, τόμος 19, σελίδες 157–167. Association for Computational Linguistics, 2019.

[25] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli και Wojciech Zaremba. *Sequence level training with recurrent neural networks*. *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[26] Zhisheng Xiao, Karsten Kreis και Arash Vahdat. *Tackling the Generative Learning Trilemma with Denoising Diffusion GANs*. *International Conference on Learning Representations*, 2021.

[27] Diederik P Kingma, Tim Salimans, Ben Poole και Jonathan Ho. *Variational Diffusion Models*. *Advances in Neural Information Processing Systems*, 2021.

[28] Alexander Quinn Nichol και Prafulla Dhariwal. *Improved denoising diffusion probabilistic models*. *International Conference on Machine Learning*, σελίδες 8162–8171. PMLR, 2021.

[29] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang και Il Chul Moon. *Soft Truncation: A Universal Training Technique of Score-based Diffusion Model for High Precision Score Estimation*. *International Conference on Machine Learning*, σελίδες 11201–11228. PMLR, 2022.

[30] Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang και Il chul Moon. *Maximum Likelihood Training of Implicit Nonlinear Diffusion Model*. *Advances in Neural Information Processing Systems*, 35:32270–32284, 2022.

[31] Tim Dockhorn, Arash Vahdat και Karsten Kreis. *Score-based generative modeling with critically-damped langevin diffusion*. *arXiv preprint arXiv:2112.07068*, 2021.

[32] Arash Vahdat, Karsten Kreis και Jan Kautz. *Score-based generative modeling in latent space*. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser και Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 10674–10685. IEEE, 2022.

[34] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho και Tim Salimans. *On distillation of guided diffusion models*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 14297–14306, 2023.

[35] Bowen Jing, Gabriele Corso, Renato Berlinghieri και Tommi Jaakkola. *Subspace diffusion generative models*. *European Conference on Computer Vision*, σελίδες 274–289. Springer, 2022.

[36] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon και Chris G Willcocks. *Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes*. *European Conference on Computer Vision*, σελίδες 170–188. Springer, 2022.

[37] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein και Kfir Aberman. *Dreambooth: Fine tuning text-to-image diffusion models for subject-driven*

generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 22500–22510, 2023.

[38] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan και Baining Guo. *Vector quantized diffusion model for text-to-image synthesis*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 10696–10706, 2022.

[39] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba και Joshua B Tenenbaum. *Compositional visual generation with composable diffusion models*. *European Conference on Computer Vision*, σελίδες 423–439. Springer, 2022.

[40] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte και Luc Van Gool. *Repaint: Inpainting using denoising diffusion probabilistic models*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 11461–11471, 2022.

[41] Hyungjin Chung, Byeongsu Sim και Jong Chul Ye. *Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 12413–12422, 2022.

[42] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut και others. *Imagen editor and editbench: Advancing and evaluating text-guided image inpainting*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 18359–18369, 2023.

[43] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri και Michal Irani. *Imagic: Text-based real image editing with diffusion models*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 6007–6017, 2023.

[44] Omri Avrahami, Dani Lischinski και Ohad Fried. *Blended diffusion for text-driven editing of natural images*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 18208–18218, 2022.

[45] Gwanghyun Kim, Taesung Kwon και Jong Chul Ye. *Diffusionclip: Text-guided diffusion models for robust image manipulation*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 2426–2435, 2022.

[46] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch και Daniel Cohen-Or. *Null-text inversion for editing real images using guided diffusion models*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 6038–6047, 2023.

[47] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa και Supasorn Suwajanakorn. *Diffusion autoencoders: Toward a meaningful and decodable representation*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 10619–10629, 2022.

[48] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon και Sungroh Yoon. *ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models*. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, σελίδες 14367–14376, 2021.

[49] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen και Baochang Zhang. *Implicit diffusion models for continuous super-resolution*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 10021–10030, 2023.

[50] Shitong Luo και Wei Hu. *Diffusion probabilistic models for 3d point cloud generation*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 2837–2845, 2021.

[51] Linqi Zhou, Yilun Du και Jiajun Wu. *3d shape generation and completion through point-voxel diffusion*. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, σελίδες 5826–5835, 2021.

[52] Zijiao Chen, Jiaxin Qing, Tiange Xiang, Wan Lin Yue και Juan Helen Zhou. *Seeing beyond the brain: Conditional diffusion model with sparse masked modeling for vision decoding*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 22710–22720, 2023.

[53] S. Kirkpatrick, C. D. Gelatt και M. P. Vecchi. *Optimization by Simulated Annealing*. *Science*, 220(4598):671–680, 1983.

[54] Radford M. Neal. *Annealed Importance Sampling*, 1998.

[55] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu και Bo Zhang. *Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models*. *International Conference on Machine Learning*, σελίδες 1555–1584. PMLR, 2022.

[56] Fan Bao, Chongxuan Li, Jun Zhu και Bo Zhang. *Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models*. *International Conference on Learning Representations*, 2021.

[57] Prafulla Dhariwal και Alexander Nichol. *Diffusion models beat gans on image synthesis*. *Advances in neural information processing systems*, 34:8780–8794, 2021.

List of Abbreviations

BCE	Binary Cross Entropy
DDIM	Denoising Diffusion Implicit Model
DDPM	Denoising Diffusion Probabilistic Model
DG	Discriminator Guidance
GAN	Generative Adversarial Network
NCSN	Noise Conditional Score Network
SDE	Stochastic Differential Equation
SGM	Score-based Generative Model
VE	Variance Exploding
VP	Variance Preserving