University of Sheffield

Learning Free Machine Learning

Michael Whealing

Supervisor: Dr Anton Ragni

A report submitted in partial fulfilment of the requirements
for the degree of MSc in Computer Science

in the

Department of Computer Science

September 8, 2025

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: Michael Whealing

Signature: M.Whealing

Date: 08/09/25

Abstract

Generative models describe a class of machine learning methods designed to learn the underlying structure of data, and produce new samples that resemble the original distribution. Contemporary works in generative modelling mainly use Score-Based Diffusion, Flow, and Energy-Based formulations. These approaches overly rely on neural parametrisations to invoke novel generation, obscuring the underlying dynamics of the process. In this work, we provide a broad unifying view of these generative models, relating them as probability transports driven by vector fields. We derive a Closed-Form Flow Model (CFFM) under this framework that specifies a time-inhomogeneous energy function with analytical gradients. Within this model, we explicitly provide control over the bias and regularisation schemes of the flows using competitive kernels. As a result, conditions for memorisation and generation can be induced, so that the underlying mechanics of the fidelity-diversity trade-off can be observed. Empirically, CFFM captures manifolds and reproduces structure; under sufficient smoothing, the energy landscape forms soft Voronoi-like tessellations where trajectories occupy barycentric intermediates from kernel overlap. This enables interpolation and novel generation, but also makes the fidelity–diversity trade-off explicit, softer tessellations increase kernel overlap, which in turn accentuates artefacts. On toy datasets and PCA image latents, temperature annealing yields plausible, diverse samples without back-propagation or parameter storage. Computationally, per sampling step, work scales linearly in dataset size under the naïve implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Generative models are a class of unsupervised and semi-supervised learning algorithms that aim to discover the hidden structure of data, without relying on external labels, or with fewer labels than previous machine learning models (Liu et al. 2017). These systems have demonstrated remarkable capabilities across a wide range of modalities, including images, speech, video, and molecule synthesis (Xu et al. 2025, Nie et al. 2025, Popov et al. 2021, Ho et al. 2022, Esser et al. 2023, Zang & Wang 2020, Shi, Xu, Zhu, Zhang, Zhang & Tang 2020). Among contemporary approaches, diffusion and score-based generative models (DMs/SGMs) have emerged as the most advanced and widely adopted techniques (Sohl-Dickstein et al. 2015, Ho et al. 2020, Song et al. 2021). Earlier paradigms, such as variational autoencoders (VAEs) (Kingma & Welling 2022), generative adversarial networks (GANs) (Goodfellow et al. 2014), normalising flows (NFs) (Papamakarios et al. 2021), and energy-based models (EBMs) (Lecun, Chopra & Hadsell 2006), are now less dominant in mainstream use. Nevertheless, the generative principles underlying these models continue to inform state-of-the-art applications and inspire methodological innovations.

Despite significant advances, practical generative systems remain both computationally demanding and conceptually opaque. Diffusion models require costly iterative inference procedures and large-scale neural parametrisations; GANs are prone to instability during training; EBMs encounter fundamental difficulties in sampling and normalisation; and normalising flows sacrifice flexibility due to strict invertibility constraints. Despite each models inherent advantages and disadvantages, the underlying generative processes that promote generalisation over memorisation are not theoretically understood (Pidstrigach 2022).

This work adopts a unifying perspective in which modern generative methods are interpreted as learning vector fields that transports a spatio-temporal probability mass in data space. DMs, SGMs and NFs are naturally framed as spatio-temporal frameworks, where the velocity field depends explicitly on both position and time. This is shown by parametrisation of the score $\nabla_x \log p_t(x)$ in DMs and SGMs (Daras et al. 2024), and the learning of a velocity field $v_t(x)$ that solves for the continuity equation in the case of Flow matching (FMs) and Continuous Normalising Flows (CNFs) (Lipman et al. 2023). In classical derivations, EBMs have only spatial interpretation, deriving a static energy proportional to a probability density, $p(x) \propto e^{-E_\theta(x)}$. However, as demonstrated, by reparametrisation of v_t as a scalar

energy function, $E(x,t)$, a time-inhomogeneity potential is recovered, whose gradient induces a velocity field. This establishes the adopted perspective in which EBMs, DMs, SGMs and Flows are instances of probability mass transport under vector fields, differing only in how the temporal dependence is parametrised and regularised. This viewpoint suggests that strategies which stabilise, regularise, or condition these fields may be transferable across model classes. This perspective reframes generative models, not as disjoint architectures, but instances of the same underlying transport problem. This lens makes it possible to transfer stabilisation, regularisation, and conditioning strategies across model classes. Furthermore, it motivates the study of learning free generative processes, in which analytically derived vector fields replace neural parametrisation. Such approaches provide interpretability, analytical tractability, and computational efficiency. In addition, by deriving analytically grounded baselines that attenuate the role of neural approximation, one can expose the generative mechanisms that are otherwise obscured by neural networks, allowing generative models to function as interpretable machines rather than black-box function approximators, thereby inciting sustainable, domain-specific generative modelling.

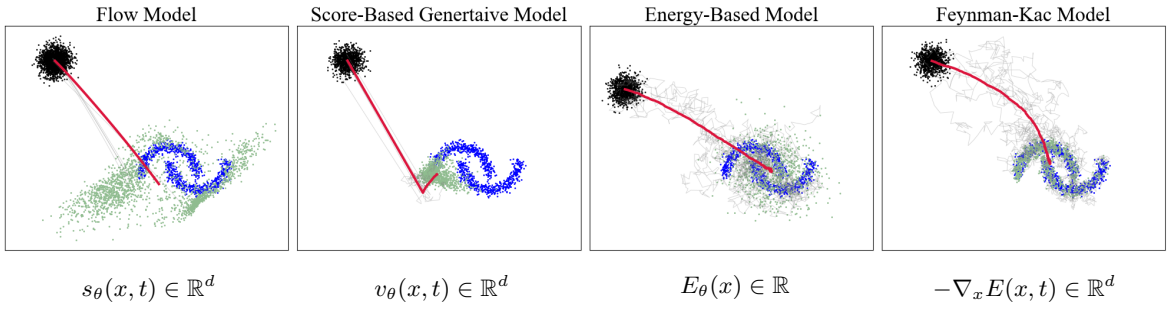


Figure 1.1: Comparative plot of generative models. Figures show the three models of study in this report and compares them to our learning-free model, CFFM. The plots demonstrate how the generated data (green) converges to the data manifold (blue) through the average trajectory (red), as well as randomly sampled individual particle trajectories (grey).

To this end, we propose an analytical framework, derived via the Feynman–Kac (FK) formulation, linking parabolic partial differential equations (PDEs) to expectations over stochastic process. This provides a non-parametric generative process with explicit control terms. Balcerak et al. (2025) show that the construction of a time-inhomogeneous energy $E(x,t)$, whose gradient yields the vector field $v_t(x) = -\nabla_x E(x,t)$, transports probability mass according to the continuity equation. By using the energy, rather than score $\nabla_x \log p_t(x)$ or free form velocity v_t, global integrability is ensured. Ensuring the score integrates to a Boltzmann density $p_t(x) \propto e^{-E(x,t)}$, aligning with the maximum entropy principle (Jaynes 1957), providing the least-biased density consistent with the constraints encoded by E. Practically, the vector field is obtained from a closed-form Monte Carlo Potential Estimator (MCPE), with explicit *terminal*, *potential*, and *source* controllers. Through these controllers, the potential encodes variance centred energy, giving the local spread and control, the terminal function supplies feature attraction, aligning the model to the data manifold, and the source provides global regularisation, preventing collapse by supplying cost over the flow. The overall effect of these

explicitly defined functions penalise off-manifold excursions along the path, and encourage controlled exploration. Analogically, the potential corresponds to variance scheduling or local smoothness regularisation in DMs and FMs, the terminal reflects endpoint alignment akin to the final score, and the source parallels entropy injection mechanisms, such as forward noise in DMs or EBM regularisation (Ho et al. 2020, Sohl-Dickstein et al. 2015, Lipman et al. 2023, Kim et al. 2023, Song & Kingma 2021). The FK decomposition into these respective terms provides explicit and interpretable controls for fidelity, local variance regulation and global regularisation respectively. However, this design does not on its own encode geometry of the manifold, or provide an adaptive learning of the energy function, delineating the limits of the framework. Because all terms are available in closed form, their gradients can be evaluated without neural networks or automatic differentiation. Sampling proceeds via Langevin dynamics that uses the MCPE, $\nabla_x \hat{E}(x,t)$.

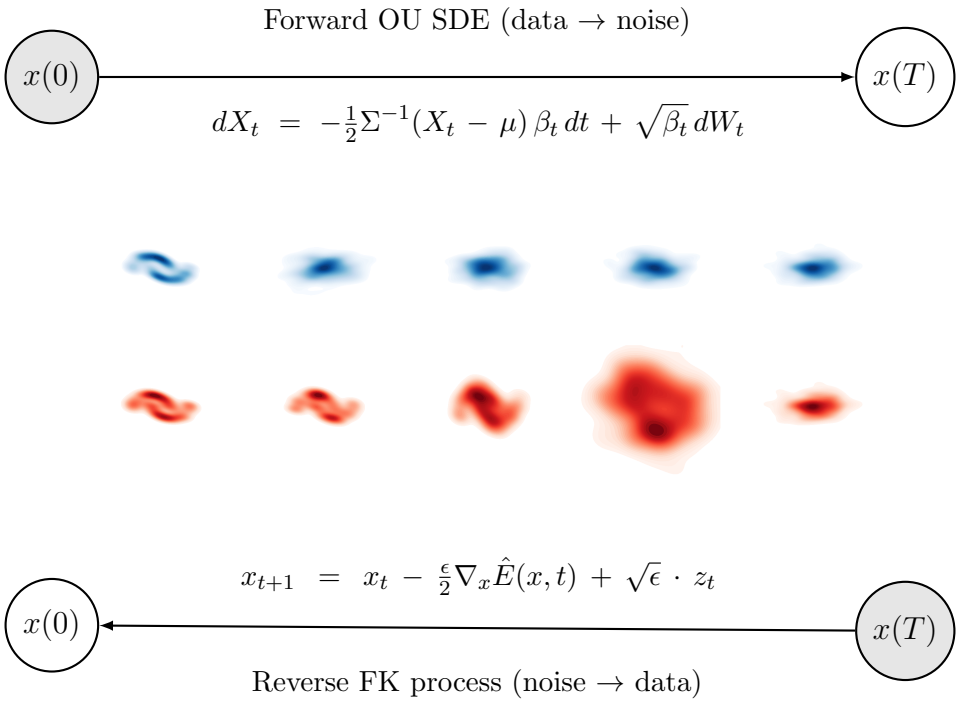


Figure 1.2: Forward Noising and Reverse Generative Process. Forward process (blue) starts from data distribution and is perturbed iteratively using the Ornstein–Uhlenbeck process. This process is reversed (red) using Langevin dynamics, where the *vector field* or (score) is described through the gradient of the scalar potential function estimated via the *GradMCPE*, $-\nabla_x \hat{E}(x,t)$

Our implementation instantiates the three FK terms using normalised log-sum-exponential (LSE) kernel-based energies. A temperature parameter rescales the kernel weighting, allowing control over the smoothness of the potential, terminal, and source functionals. A forward Ornstein–Uhlenbeck (OU) schedule supplies a time-dependent attenuation profile, naturally shaping a progression from broad exploration at early times to refinement near the terminal distribution. The resulting sampler is light-weight and interpretable, whereby modifying a single temperature or kernel bandwidth produces predictable adjustments in the induced

vector field, directly controlling the trade-off between sample sharpness and diversity.

Empirically, on canonical two-dimensional manifolds we visualise the induced vector fields and show how terminal and source temperatures trade sharpness for coverage. On image datasets, with application of principal component analysis as the dimensionality reduction technique, the method produces recognisable yet smooth samples. We also report ablations on entropy of path weights and effective neighbourhood size, connecting the temperature and kernel choices to controllable generalisation behaviour.

The contributions formulated in this report are threefold. (i) Firstly, a concise vector-field framing that places diffusion/score models, EBMs, and Flows under a common transport formalism, clarifying similarities and differences in their objectives. While related unifications have been noted in literature, our presentation emphasises the probability transport as the common thread. (ii) Secondly, we derive a closed-form FK-based reverse process and its MC potential estimator with explicit terminal, potential, and source components enabling learning-free generation with transparent inductive biases. Although similar processes exist in the form of optimal transport theory, particularly schrodinger bridges (Tong et al. 2024), this approach provides non-parametric formulations. (iii) Thirdly, we implement an efficient sampler, using Langevin dynamics, demonstrating on toy and image domains how kernel and temperature parameters govern the sharpness–diversity trade-off. In this framework, this trade-off reduces to adjusting the temperature parameter within the kernel weighting, giving a direct and predictable effect on the sampler's behaviour. Unlike parametric generative models, where sharpness and diversity is an emergent property of the training dynamics, here it is analytically transparent, low temperatures concentrate mass, while higher temperatures diffuse mass. This provides an interpretable generative model that foregrounds analytical structure over large-scale parametrisation.

Chapter 2

Preliminaries

2.1 Energy Based Models

A core aim of statistical modelling is to capture dependencies between observed and latent variables, enabling reliable inference from partial information (LeCun, Chopra, Hadsell, Ranzato, Huang et al. 2006). When only limited constraints are known, most-notably the expected energy, $\langle E \rangle$, the choice of Probability Density Function (PDF) $p(x)$, to adopt without introducing unjustified assumptions is paramount. Jaynes' Maximum-Entropy Principle (Jaynes 1957) demonstrated that among all distributions, subject to the constraints, the distribution that maximises Shannon entropy is optimal. In the case where only the mean is constrained, the distribution is exponential, if the mean and variance is known, the distribution yields a Gaussian. For the scenario where the expected energy $\langle E \rangle$ of an observable is constrained, the Maximum-Entropy solution follows a Boltzmann distribution. This formalism through the expected energy does not require full knowledge of the energy function, only that the energy is the constrained quantity expectation. Formally, for a given variable x and an energy function $E(x)$, the Maximum-Entropy problem is expressed by,

$$\begin{aligned} \max_{p(x)} & \int -p(x)\ \log p(x)\, dx \\ \text{s.t.} & \int p(x)E(x)\, dx = \langle E \rangle, \\ & \int p(x)\, dx = 1 \end{aligned} \tag{2.1}$$

The distribution satisfied above is determined to be the Boltzmann (or Gibbs) distribution,

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta} \qquad Z_\theta = \int e^{-E_\theta(x)}\, dx \tag{2.2}$$

This formulation gives rise to the concept of *Energy-Based Models* (EBMs). For which the probability distribution is defined implicitly through a non-linear regression function (or

energy function) $E_\theta(x)$, with parameters θ (Huembeli et al. 2022). In practice, it is possible to parametrise the energy directly or its gradient, $-\nabla_x E_\theta(x)$. This relationship indicates probable distributions exist in regions of low energy. The normalising constant, typically named the partition function Z_θ, ensures the distribution is normalised over all state-space, though it is often intractable in practice.

EBMs are commonly described in probabilistic terms as *Globally Normalised Models*, where the probability of a configuration is defined through an energy function. The energy is typically factorised into additive contributions over cliques or local subsets of variables (an energy-based factorisation) so that the joint distribution is globally normalised but locally structured (Koller & Friedman 2009). More generally EBMs can be defined as an un-normalised score through the energy function, or controversially a diffusion (Song et al. 2021, Pidstrigach 2022), that rely on the EBMs partition function to convert this into a valid probability distribution. This formulation grants EBMs several properties that make them appealing for statistical modelling. The key advantages include (Ou 2024),

1. **Flexible parametrisation:** Since the partition function Z_θ is not computed explicitly during training, the energy function can be parametrised in expressive forms. The absence of a strict normalisation requirement during training enables highly expressive energy functions (e.g. deep neural networks).
2. **Bypass of the partition function:** objectives such as score matching or contrastive divergence allow EBMs to be trained without evaluating Z_θ.
3. **Unified modelling:** SGMS, DMs and FMs are primarily generative, whereas EBMs naturally support hybrid generative–discriminative and semi-supervised learning setups within the same framework.

The primary challenge in using EBMs lies in the intractability of the partition function, which is required to compute exact probabilities. However, this caveat can be circumvented using alternative training and inference objectives. For training EBMs, score matching (Lyu 2012), noise-contrastive estimation (Gutmann & Hyvärinen 2010), and contrastive divergence (Hinton 2002) are commonly employed, which avoid the need to compute or differentiate Z_θ directly.

For contextualisation of these approaches, the standard method for fitting probabilistic models from i.i.d. data is via Maximum Likelihood Estimation (MLE). Given dataset $\mathcal{X}$, we can frame $p_\theta(x)$, where $x \in \mathcal{X}$, as the model that is parametrised by θ, and $p_{data}(x)$ as the true underlying data distribution of dataset. MLE aims to find parameters θ that maximise the likelihood of the observed model, specifically $\theta^* = \underset{\theta}{\operatorname{argmax}} \prod_{x \in \mathcal{X}} p_\theta(x)$. In practice, it is more convenient and numerically stable to minimise the expectation of the Negative Log-Likelihood (NLL), leading to the equivalent objective function (Song & Kingma 2021),

$$\theta^* = \underset{\theta}{\operatorname{argmin}} - \mathbb{E}_{x \sim p_{data}(x)}[\log p_\theta(\mathbf{x})] \tag{2.3}$$

This objective measures how well $p_\theta(x)$ approximations the distribution over the dataset $p_{data}(x)$. Equation 2.3 forms the basis of learning EBMs and many generative approaches.

Minimising the NLL can again also be interpreted as minimising the Kullback-Leibler (KL) divergence between distributions $p_\theta(x)$ and $p_{data}(x)$. Expanding the expectation in Equation 2.3 we achieve (see appendix section A.1.1 for derivation),

$$\begin{aligned} -\mathbb{E}_{x\sim p_{\text{data}}(\mathbf{x})}[\log p_\theta(x)] &= D_{\text{KL}}(p_{\text{data}}(x) \,\|\, p_\theta(x)) - \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log p_{\text{data}}(x)] \\ &= D_{\text{KL}}(p_{\text{data}}(x) \,\|\, p_\theta(x)) + \text{const} \end{aligned} \tag{2.4}$$

The NLL of the EBM cannot be evaluated directly due to the intractability of the partition function Z_θ. In optimisation, however, it is the gradient of the NLL that is required. This gradient can be expressed in terms of the energy function and the partition function seen in Equation 2.2, and thus provides a principled update direction for parameter learning. In practice, approximating the expectation over the model distribution often requires Monte Carlo methods such as Langevin dynamics (Younes 1999), but the formulation of the gradient itself does not depend on the specific sampling scheme used,

$$-\nabla_\theta \log p_\theta(x) = \nabla_\theta E_\theta(x) + \nabla_\theta \log Z_\theta. \tag{2.5}$$

Using the log-derivative trick,

$$\nabla_\theta \log Z_\theta = \frac{1}{Z_\theta}\nabla_\theta Z_\theta = -\,\mathbb{E}_{x'\sim p_\theta}\big[\nabla_\theta E_\theta(x')\big],$$

we obtain

$$-\nabla_\theta \log p_\theta(x) = \nabla_\theta E_\theta(x) - \mathbb{E}_{x'\sim p_\theta}\big[\nabla_\theta E_\theta(x')\big]. \tag{2.6}$$

This decomposition shows that decreasing the energy on data points (first term) must be balanced by the parameter-dependence of the partition function (second term), which effectively increases energy on configurations sampled from the current model. While $\nabla_\theta E_\theta(x)$ is directly available via automatic differentiation, the expectation $\mathbb{E}_{x'\sim p_\theta}[\nabla_\theta E_\theta(x')]$ is intractable and must be approximated though sampling methods (see Appendix A.1.2 for full derivation). If approximate samples from the model can be drawn, stochastic gradient descent (SGD) can be applied using Monte Carlo estimates of this expectation, yielding an unbiased gradient estimator for the NLL (Song & Kingma 2021).

While EBMs were historically considered distinct from other generative methods, it can be shown that contemporary generative models can be reinterpreted as energy-based or score-based methods under different formulations. These models shift focus from explicitly learning the normalised density to learning auxiliary functions, such as score functions, transport maps, or de-noising processes, that implicitly define the target distribution. Most Notably; Score-Based Generative Models (SGMs)(Song et al. 2021), which learn the score function directly; De-Noising Diffusion Probabilistic Models (DDPMs) (Ho et al. 2020), which perform modelling via time-reversible de-noising; and more recently a unified approach to energy flow-matching (Balcerak et al. 2025), which learn velocity fields that match optimal transport paths between base and target distributions.

While EBMs were historically considered distinct from other generative methods, many contemporary approaches can be reinterpreted as energy, or score-based models under alternative

formulations. Instead of learning the normalised density directly, these methods learn auxiliary objectives in the form of the score function, transport maps, or denoising processes, that implicitly define the distribution. For example, Score-Based Generative Models (SGMs) (Song et al. 2021) learn the score function $\nabla_x \log p(x)$ explicitly; Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al. 2020) parametrise time-reversible denoising processes; and recent energy flow-matching approaches (Balcerak et al. 2025) learn velocity fields aligned with optimal transport paths between base and target distributions.

In the following section, we briefly explore these connections and show how they can all be viewed as instances of a broader energy-based modelling framework.

2.1.1 Energy-Based Model Training

For training of EBMs, Score Matching (Hyvärinen 2005) (SM) offers an alternate MLE objective, by minimising the Fisher divergence between $p_{data}(x)$ and $p_\theta(x)$ (Song et al. 2020). Unlike other likelihood methods, SM bypasses the intractability of Z_θ by matching the gradient log density (or scores) w.r.t. the data. This leads to the objective function expression (see appendix section A.1.3 for derivation),

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\tfrac{1}{2}\|\nabla_x E_\theta(x)\|^2 - \Delta_x E_\theta(x)\right], \tag{2.7}$$

Where $\Delta_x E_\theta(x)$ represents the Laplacian of $E_\theta(x)$. Equation 2.7 represents the basis objective function used for training EBMs. A useful redefinition of this objective function is described through the Fisher Divergence, wherein for training EBMs we first transform the equivalence of distributions to the equivalent score definitions $-\nabla_x E_\theta(x) = \nabla_x \log p_\theta(x)$,

$$D_F(p_{\text{data}}(x) \,\|\, p_\theta(x)) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\tfrac{1}{2}\|\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x)\|^2\right]. \tag{2.8}$$

Through this framing, the identity $-\nabla_x E_\theta(x) = \nabla_x \log p_\theta(x)$ makes explicit the equivalence between the energy gradient and the model score (Song & Kingma 2021). However, this condition only holds when the score is continuously differentiable and finite everywhere. In realistic applications, these assumptions are rarely satisfied. For example, in image generation the data lies in a bounded domain ($[0, 255]^d$), creating discontinuities at the boundaries where the true score would diverge to $-\infty$. More generally, neural networks trained via score matching objectives are not constrained to produce curl-free vector fields, and thus may yield approximate scores that are not globally integrable into a valid density (Chen, Huang, Zhao & Wang 2023). A common remedy is to add noise to the data, perturbing each point as $\hat{x} = x + \varepsilon$, where the noise distribution $p(\varepsilon)$ is smooth. This denoising score matching regularises the objective of Equation 2.8, smoothing the empirical distribution and ensuring the conditions for score matching are approximately met (Kingma & LeCun 2010). Nevertheless, the approach requires costly second-order derivatives and still produces score fields that are only approximate, rather than guaranteed *true* gradients of an underlying density.

2.1.2 Inference and Sampling in Energy-Based Models

The inference process in EBMs requires generating samples from the model distribution, $p(x) \propto e^{-E(x)}$. Since EBMs model a physical systems, which naturally evolves toward equilibrium, they follow complex stochastic evolution laws (Bösch et al. 2025). Langevin dynamics can produce samples from a probability density $p(x)$ using only the score function, $\nabla_x \log p(x)$. Given step size $\epsilon > 0$, and $x_0 \sim \pi(x)$, where π is the prior distribution at $t = 0$, the Langevin method recursively computes the following (Song & Ermon 2020) (see Appendix Section 1.1.4 for detail),

$$x_t = x_{t-1} + \frac{\epsilon}{2}\nabla_x \log p_\theta(x_{t-1}) + \sqrt{\epsilon}\, z_t \tag{2.9}$$

where $z_t \sim \mathcal{N}(0, I)$ is Gaussian noise. This iterative scheme produces approximate samples from $p_\theta(x)$, enabling sampling from EBMs without requiring evaluation of the partition function.

Note that sampling from Equation 2.8 only requires only $\nabla_x \log p_\theta(x_{t-1})$. Implying, to obtain samples from $p_{data}(x)$, a network can be trained to approximate the score where $\nabla_x \log p_\theta(x_{t-1}) \approx \nabla_x \log p_{data}(x)$, and then obtain approximate samples with Langevin dynamics (Song & Ermon 2020). This principle of sampling via Langevin dynamics informs the framework for SGMs, that fit into the broader class of *Diffusion Models*.

2.2 Diffusion and Flows

Diffusion models (DMs) derive from non-equilibrium statistical physics (Sohl-Dickstein et al. 2015). Essentially, DMs are a class of probabilistic models that progressively perturb the initial data with noise, then learn the reverse process for sample generation (Yang, Zhang, Song, Hong, Xu, Zhao, Zhang, Cui & Yang 2024). Generally DMs, can be formulated under three primary classes, denoising diffusion probabilistic models (DDPMs) (Ho et al. 2020), stochastic differential equations (Song et al. 2021), and mentioned score matching generative models (SGMs) Song & Ermon (2020).

Diffusion Models

Diffusion Models (DMs) can be formulated either in discrete or continuous time. Discrete-time formulations use parametrised Markov chains, trained via variational inference, to gradually transform noise into data samples (Sohl-Dickstein et al. 2015, Ho et al. 2020). In contrast, continuous-time formulations describe the forward noising and reverse denoising dynamics through stochastic differential equations (SDEs), providing a unifying and more flexible view of the generative process. In the forward process, the SDE smoothly transforms the distribution $p_{data}(x)$ toward a known prior $p(x)$ through injecting noise ε. From this noise, the reverse process solves the reverse stochastic differential equation (reverse SDE) (Pardoux & Peng 1992), recovering the data. Importantly, this reverse SDE only depends on the time-dependent gradient field (score) of the perturbed data distribution (Song et al. 2021). This framework encapsulates score-based (Hyvärinen 2005) and diffusion probabilistic modelling.

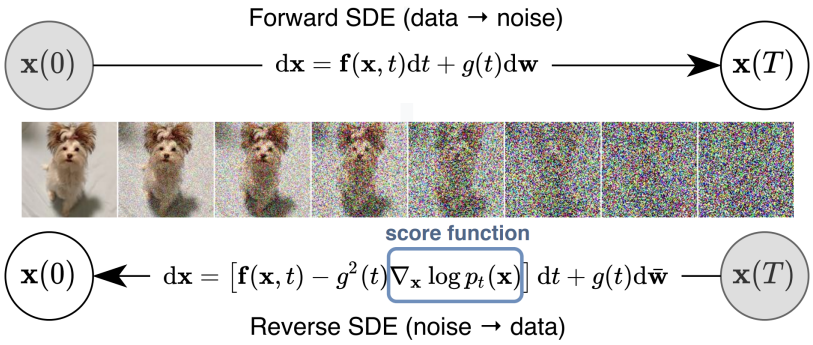


Figure 2.1: *Overview of the forward and reverse SDE process (Song et al. 2021),*

The goal of the forward process is to construct the diffusion process so that $\{x(t)\}_{t=0}^{T}$ evolving over continuous time $t \in [0, T]$, such that the data distribution $x_0 \sim p_{data}(x)$, for which a dataset of i.i.d samples, and the prior distribution $x(T) \sim p_T(x)$ exist. Under these conditions, there is a tractable process to generate samples efficiently. Firstly, to perturb the data, a forward SDE is defined using the Itô diffusion process, where X_t defines the stochastic process evolving under diffusion dynamics,

$$dX_t = f(X_t, t)dt + g(t)\, dW_t \tag{2.10}$$

Where functions $f(X_t, t)$ and $g(t)$ define the time dependent vector flow field of the deterministic evolution (drift), and set of vector fields that define the coupling of the system to Gaussian noise $\mathcal{N}(0, I)$ (diffusion coefficient) respectively, as well as W_t describing the Weiner process (Brownian motion). At the end of the forward diffusion, the system reaches a prior distribution $p_T(x)$, typically a standard Gaussian. Initialising the reverse process with $x_T \sim p_T(x)$, the reverse SDE recovers samples $x_0 \sim p_{data}(x)$ (Anderson 1982),

$$dX_t = \left[f(X_t, t) - g(t)^2 \nabla_x \log p_t(x)\right] dt + g(t) d\overline{W}_t \tag{2.11}$$

Where $\overline{W_t}$ denotes the standard Wiener process as time flow from $T \to 0$, and dt now provides the negative infinitesimal time step. Noticing $\nabla_x \log p_t(x)$ is the score of each marginal distribution, by deriving the reverse diffusion process from Equation 2.11, the initial distribution $p_{data}(x)$ can be recovered by flowing $p_T(x) \to p_{data}(x)$.

While the reverse SDE in Equation 2.11 provides one path to sample from the data distribution, there exists an equivalent deterministic process that generates samples from the same marginal distributions (Song et al. 2021). This deterministic counterpart, known as the probability flow ODE, offers several computational advantages and forms a crucial bridge to understanding flow-based models.

$$dX_t = \left[f(X_t, t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)\right] dt \qquad t : T \to 0 \tag{2.12}$$

This formulation is obtained by removing the stochastic term of the reverse SDE while preserving the same marginal distributions $p_t(x)$ at each time t (Chen et al. 2019). Retaining the marginals ensures theoretical equivalence to the stochastic process, but yields a tractable deterministic formulation in the form of the probability flow ODE. Although the scores $\nabla_x \log p_t(x)$ must be known or approximated for the ODE to be applied, preserving the marginals confers several advantages that make this formulation particularly useful,

1. **Bijective Mapping**: Unlike the reverse SDE, the probability flow ODE defines a bijection between the noise distribution $p_T(x)$, and the data distribution $p_{data}(x)$, under standard existence and uniqueness conditions (Øksendal 2003).

2. **Exact Likelihood**: Due to the transformation being deterministic and supporting a bijective, the exact likelihood can be computed using the instantaneous change of variables,
$$\log p_0(x_0) = \log p_T(x_T) - \int_0^T \nabla \cdot v_t(x_t)\, dt$$
where the velocity vector,
$$v_t(x_t) = f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)$$
which is the vector field of the ODE.

3. **Faster Sampling**: Applying the deterministic ODE in the reverse process provides samples of high quality, with fewer sampling steps, as sampling follows a predictable path from noise to data due to the elimination of random noise injections (Chen, Chewi, Lee, Li, Lu & Salim 2023).

Crucially, the probability flow ODE can be reinterpreted as learning a time-dependent vector field, $v_t(x)$, that transport the probability mass from the prior $p_T(x)$, to the data distribution $p_{data}(x)$. This vector field specifies the *flow* of the density over time, where at each position x and time t the vector field $v_t(x)$ encodes both the direction and magnitude of transport. The divergence, $\nabla_x \cdot v_t(x)$, quantifies how this density locally concentrates or disperses. Alternatively, integrating the trajectories of $v_t(x)$ yields sample paths, tracing the evolution of individual points during generation.

This vector field interpretation connects SGMs, ensuring that the evolving distributions p_t interpolate between prior and data, as defined through the probability flow ODE and linked to optimal transport theory. Importantly, this guarantee holds at the level of marginals; the full joint law over trajectories is not uniquely determined, with SGMs selecting a stochastic path measure and NFs a deterministic coupling.

While SGMs learn the score function $\nabla_x \log p_t(x)$ and derive the vector field, if instead the vector field itself is parametrised and learnt, Flow Matching emerges. This reframing motivates Flow Matching and Continuous Normalising Flows (CNFs) (Chen et al. 2019), by parametrising the vector field via neural networks, $v_\theta(x_t)$. The model then learns to match the true transport vector field, bypassing the need to first learn the scores before deriving the flow. Through this framework, CNFs are able to learn the density function by evolving simple distributions such as $\mathcal{N}(0, I)$ toward complex distributions (Gao et al. 2024, Kobyzev et al. 2021).

The relationship between densities, vector fields, and paths is as follows. The *density path*, $\{p_t(x)\}_{t\in[0,T]}$, describes how probability distributions evolve in time, interpolating between a prior p_T and the data distribution p_{data}. The *vector field*, $v_t(x)$, specifies the instantaneous direction and speed with which probability mass is transported at location x and time t. Finally, by integrating $v_t(x)$ forward in time, one obtains *sample paths*, i.e. trajectories $\{X_t\}$ of individual particles that together realise the evolution of the density. In this way, densities capture the population statistics, vector fields describe the underlying transport mechanism, and paths provide the instance-level trajectories consistent with both.

The key insight is that both SGMs, and CNFs solve the same fundamental problem in how to parametrise the vector field that optimally transports the probability mass from a simple prior, toward the complex data distribution. Given a target probability density path $p_t(x)$, and its corresponding vector field $v_t^*(x)$, the Flow Matching objective is defined as,

$$\mathcal{J}_{FM}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),x\sim p_t(x)}||v_t(x) - v_t^*(x)||^2 \tag{2.13}$$

where θ denotes the learnable parameters of the CNF vector field v_t, $t \sim \mathcal{U}(0,1)$, and $x \sim p_t(x)$. More simply, the neural network v_θ, is trained to approximates the vector field v_t^* (Lipman et al. 2023). Whilst initially appealing, the Flow Matching approach is intractable under naïve assumptions, as p_t and v_t^* are unknown. To overcome this, the velocity field can be redefined as a parametrisation of a scalar energy function. By restricting the velocity vector v_t, that is determined through the probability flow ODE, the gradient of a scalar potential is restricted, since any differentiable scalar field $E(x,t)$ induces a vector field via its gradients,

$$v_t(x) = -\nabla_x E_\theta(x,t) \tag{2.14}$$

This restriction induces conservative flows derived from an energy landscape, consistent with the EBM framework (Balcerak et al. 2025). Training then reduces to adjusting E_θ, such that its induced gradients approximate the target transport field, $v_t^*(x)$, that satisfies the continuity equation,

$$\partial_t\, p_t(x) + \nabla_x \cdot (v_t^*(x) p_t(x)) = 0 \tag{2.15}$$

Thus, the choice of potential $E_\theta(x,t)$ is admissible, provided its induced vector field $v_t(x) = -\nabla_x E_\theta(x,t)$ is conservative and transports probability mass in a way that satisfies the continuity equation. This allows the velocity to be conserved, as it comes from the energy landscape. By substituting this into Equation 2.13, the objective function becomes,

$$\mathcal{J}_{FM}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),x\sim p_t(x)}|| - \nabla_x E_\theta(x,t) - v_t^*(x)||^2 \tag{2.16}$$

In this view, we directly train the energy so that its gradients reproduce the correct flow field (Balcerak et al. 2025). A subtle but important consequence of enforcing $v_t = -\nabla_x E(x,t)$ is its connection to entropy. Conservative flows are curl-free and globally integrable, so probability

transport can always be interpreted as sliding "downhill" on an energy landscape. Moreover, under the Fokker–Planck formulation, gradient drift corresponds to the steepest descent of a free-energy functional (Jordan et al. 1997),

$$\mathcal{F}(p) = \int V(x)p(x)dx + \frac{1}{2}g(t)^2 \int p(x)\log p(x)dx \tag{2.17}$$

which combines expected energy with (negative) entropy, weighted by the diffusion strength $g(t)^2/2$. Along the flow, $\mathcal{F}[p_t]$ decreases monotonically, so entropy regularisation emerges naturally from the conservative structure.

2.3 Unifying View

The implication of these formulations is that the apparent distinctions between Diffusion Models, Energy-Based Models, and Flow Matching approaches may be artefacts of parametrisation rather than fundamental differences. By treating a scalar quantity as a time-dependent flow, the proposed framework collapses these categories into a single generative paradigm. Scalars need not remain static as any scalar potential $E(x)$ can be endowed with explicit time inhomogeneity, yielding $E(x,t)$. Under this construction, EBMs that are traditionally defined through a static energy function, provide the clearest case study. Once made time-dependent, as shown in Equation 2.14, the scalar induces a flow

$$v_t(x) = -\nabla_x E(x,t),$$

transforming EBMs from static scalar descriptions into dynamic, time-evolving flows. This reinterpretation demonstrates that EBMs are not fundamentally distinct from diffusion or flow-based models, but instead represent a specific parametrisation regime within a broader unified framework of generative modelling.

Contemporary works already reflect fragments of this unifying perspective through different scalar choices, most notably SGMs encode the variance schedule as a scalar (Song et al. 2021), energy-based flows use the Boltzmann energy $E(x)$ (Balcerak et al. 2025), and flow matching adopts a transport potential scalar (Lipman et al. 2023). However, what remains absent is a formal descriptor of the generative framework that accommodates all these adaptations. In summary, generative models share the same underlying structure in that they differ only in how the scalar potential is defined and subsequently parametrised into a vector field,

$$\text{Scalar Potential } E(x,t) \xrightarrow{-\nabla_x} v_t(x) \xrightarrow{\text{flow}} p_t(x) \tag{2.18}$$

This mapping formalises the scalar-to-vector unification view, where differences across model classes reduce to distinct parametrisations of the underlying scalar.

Chapter 3

Related Work

State-of-the-art (SOTA) generative models achieve remarkable performance by combining extensive parametrisation with modern neural architectures. When network capacity and training data are sufficiently scaled, these models are able to capture and generate increasingly high-dimensional structures (Shi, Zhou, Qiu & Zhu 2020, Labs et al. 2025, Ju et al. 2024, Google DeepMind 2025, Jumper et al. 2021). However, due to their training regimes, they require long training times and excessive computation (Wang et al. 2023, Geng et al. 2024). While recent works explore distillation, discretisation, and likelihood-based objectives to mitigate these costs (Salimans & Ho 2022, Fang et al. 2025, Kim et al. 2022), SOTA models still rely on resource-intensive parametrisations (Ma et al. 2024).

When appropriately tuned, generative models exhibit strong generation capabilities, but if poorly tuned they risk reproducing the training data, resulting in simple memorisation rather than genuine generation. This memorisation is an unwanted feature for a myriad of reasons, some of the most prominent being breach of copyright law, or leaking of private data stemming from the training data (Ross et al. 2025). Understanding the conditions under which memorisation occurs is therefore fundamental to creating reliable models. The most widely adopted generative models in current SOTA deployment are diffusion, flow and energy-based models. As shown in the Preliminaries section 2.2, these models share a similar underlying structure, where they can be broadly generalised to vector field learners, differing only in the type of scalar field that is parametrised. For example, in diffusion models, the variance schedule scalar with time dependence, forms the score field $\nabla_x \log p_t(x)$ (Song et al. 2021); in energy-based models, a scalar energy $E(x)$ defines the field through its gradients $v_t(x) = -\nabla_x E(x)$ (Balcerak et al. 2025); and in flow-based models, a scalar transport potential $\phi(x, t)$ (Lipman et al. 2023) generates the velocity field $v_t(x) = -\nabla_x \phi(x, t)$.

While the unified vector field view aids in formulating these models under similar notation and relative behaviour, current research tends to lack explanation of the mechanics for how these vector fields effect the generative process, what conditions the fields are under for novel generation, and when they memorise. Current methods parametrise their flows via neural networks, the effect of which essentially makes the theoretical understanding of their generative behaviour opaque by relying on expensive black box parametric approximations (Chang et al. 2018).

Recent theoretical analysis suggests that effective generalisation in generative models arises from inductive biases rather than sheer parametric capacity. This is supported by findings that diffusion models produce near-identical samples regardless of architecture or parametrisation (Niedoba et al. 2025, Zhang et al. 2024), implying that their behaviour is governed by structural biases rather than model scale. Score-based models encode such biases through their score functions, $\nabla_x \log p_t(x)$, which direct trajectories toward data manifolds (Pidstrigach 2022), while diffusion models exhibit a Gaussian bias shaped by the empirical mean and covariance of the training data (Li et al. 2024). This bias is most pronounced under limited capacity or early stopping, where generalisation emerges from learning low-dimensional structures instead of memorising high-dimensional samples (Niedoba et al. 2025). A related phenomenon has been identified in the context of hallucinations, diffusion models appear to generate novel samples by smoothly interpolating between nearby modes in the training set (Aithal et al. 2024). Taken together, these findings highlight how generalisation and even apparent creativity can be traced back to model biases, yet the opacity of neural parametrisations continues to obscure the precise mechanisms, leaving open the challenge of developing learning-free formulations with explicit control.

One approach to constructing a learning-free model, given a finite dataset, is to compute the score function $\nabla_x \log p_t(x)$ in closed form. However, this yields only memorisation of the training data, since the score points exactly to the data samples. Building on prior theoretical work showing that neural SGMs generalise through approximation error (Pidstrigach 2022, Yi et al. 2023, Aithal et al. 2024), Scarvelis et al. (2025) demonstrate this effect explicitly by instantiating it in closed form. They show that because neural networks cannot represent the exact score, their approximation error smooths the score field so that it points toward the barycentres of training samples rather than the samples themselves. This smoothing provides a principled explanation for why neural SGMs generalise beyond memorisation, whereas closed-form models that compute the exact score do not.

Theoretical and empirical analyses of generative processes have focused primarily on diffusion models, with comparatively less systematic investigation into the generative behaviour of energy-based and flow models. EBMs offer flexibility and an explicit, interpretable energy landscape $p(x) \propto e^{-E_\theta(x)}$ (Song & Kingma 2021, Yu et al. 2020), whereas flows provide continuous, likelihood-tractable mappings (Chen, Chewi, Lee, Li, Lu & Salim 2023, Lipman et al. 2024). Despite receiving less attention, these models can be related to diffusion models under the probability flow ODE, where score-based models form a subclass of vector-field generative models (Song et al. 2021, Lipman et al. 2023). This unifying view suggests that if score-based models achieve generalisation through approximation-induced smoothing toward data barycentres (Pidstrigach 2022, Yi et al. 2023, Aithal et al. 2024), then energy-based models learning scores $-\nabla_x E_\theta(x)$ and flow-based models learning velocity fields $v_\theta(x, t)$ may exhibit analogous behaviour. When these fields are specified analytically rather than via neural parametrisation, their dynamics can be directly inspected, offering a path toward more transparent generalisation mechanisms. Yet the black-box nature of neural parametrisations obscures whether and how these mechanisms manifest across different vector field representations (Chang et al. 2018). This limitation prevents a unified understanding of generative model behaviour and hinders the development of controllable, interpretable approaches that could explicitly leverage such approximation effects. The absence of learning-free, analyti-

cally tractable methods that can both demonstrate and regulate these mechanisms represents a fundamental gap in the understanding of how vector-field-based generative models balance memorisation and generalisation. A learning-free approach, that offers explicit control over this vector-field offers a probe into these processes. By introducing direct control into the forming of the energy landscape, the flows can be regularised to conform to certain behaviours. In this work, we introduce these control dynamics via the Feynman-Kac formula.

The Feynman–Kac (FK) formula offers a practical method to design stochastic flows, as it establishes the connection between linear PDEs and expectations of stochastic processes under Brownian motion. This provides a way to solve linear PDEs via Monte Carlo (MC) simulations of random trajectories (Pham 2014). By extending the FK formula to fully non-linear PDEs, a probabilistic representation in terms of backward stochastic differential equations (BSDEs) is obtained (Lin 2012). In the context of generative modelling, the reverse-time SDE, central to score-based models, can be viewed as a particular instance of such a representation, where its drift admits a closed-form expression in terms of the time-dependent score function, $s_\theta(x,t) = \nabla_x \log p_t(x)$ (Deveney et al. 2023). The FK formula and the related Partial Differential Integral Equation (PDIE) also have important applications in financial mathematics, where they provide an analogue of the Black–Scholes PDE for option pricing in multidimensional Lévy markets (Lin 2012). Although not carried out in this study, possibilities for jump-diffusion generative models that directly transition between distant modes while maintaining analytical tractability may stem from this formulation.

Physically, the FK formula also arises in quantum mechanics and thermodynamics, where it provides a probabilistic interpretation of path integrals and diffusion. In this work, we relate the PDE solution $u(x,t)$ to the FK solution and define a time-inhomogeneous energy $E(x,t) = -\log u(x,t)$,

$$-\nabla_x E(x,t) = \nabla_x \log u(x,t).$$

When the FK construction is chosen such that $u(x,t) \propto p_t(x)$, this recovers the data score $\nabla_x \log p_t(x)$. This formulation enables explicit control over the generative vector field, since adjusting the analytic form of $E(x,t)$ reshapes the landscape and thereby biases the induced flow.

Chapter 4

Methodology

4.1 Notation Summary Table

Table 4.1: *Summary of notation used throughout the methodology*

Symbol	Description
$X_t = \{x_l\}_{l=1}^{N} \in \mathbb{R}^D$	denotes the empirical dataset sampled from p_t
t	Global system time (current time of evaluation)
T	Final time horizon of the simulation
k	The batch element for each sample
s_j	Discretised time steps along a stochastic trajectory, from $t^{(k)}$ to T
Δs	Discrete step size between s_j and s_{j+1}
$x^{(k)}$	Initial condition for the k-th sample (batch element)
$X_{s_j}^{(i)}$	State of the i-th Monte Carlo path for sample k at time s_j
$V(X_{s_j}, s_j)$	Potential energy function evaluated at state X_s
$g(X_\tau, \tau)$	Source function evaluated for an arbitrary time τ
$f(X_T)$	Terminal function evaluated at time $t = T$ during the reverse process
$\phi^{(i)}$	Potential weighting of the i-th trajectory for a sample, used in MC estimation
$\hat{E}(x^{(i)}, t^{(i)})$	Estimated energy via Feynman-Kac expectation from $x^{(i)}$ at time $t^{(i)}$
M	Number of Monte Carlo trajectories sampled per initial condition
K	Number of discretised time steps from $t^{(k)}$ to T

Within this section we present the theory and derivation of the proposed model. We introduce a Closed-Form Flow Model (CFFM), in which the vector field is expressed analytically as the gradient of a scalar potential $E(x,t)$. To situate this model in the generative regime, we first define the forward noising process, followed by the derivation of the reverse process, and the specification of the kernel choices for the terminal, potential, and source functions.

4.2 Forward Process

Contemporary Score-Based Diffusion Models are able to approximate trajectories of stochastic processes, thereby satisfying given Stochastic Differential Equations (SDEs) through the use of Markov Chains (Song et al. 2021). In this study, the model is defined in terms of SDEs, rather than Markov Chains. This allows for the application of a forward process that transform the data distribution into noise through a fixed, data-agnostic process. Firstly, to define the forward diffusion process, an Ornstein–Uhlenbeck (OU) process SDE is formed so that as $t \to \infty$, the initial data distribution $p_{data}(x)$ is perturbed toward Gaussian noise $\mathcal{N}(\mu, \Sigma)$ that matches the data's natural scale and location (Popov et al. 2021),

$$dX_t = -\frac{1}{2}\Sigma^{-1}(X_t - \mu)\beta_t\, dt + \sqrt{\beta_t} dW_t \qquad t \in [0, T] \tag{4.1}$$

Variable Σ^{-1} defines the inverse covariance of the distribution, vector μ gives the mean of the initial data distribution, and the non-negative function β_t provides the noise schedule. These parameters define a mean-reverting drift that guides trajectories toward the data's natural location and scale. The solution for the given SDE is then given by,

$$X_t = X_0\, e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} + \left(I - e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s}\right)\mu + \int_0^t \sqrt{\beta_s}\cdot e^{-\frac{1}{2}\Sigma^{-1}(\Lambda_t - \Lambda_s)} dW_s \tag{4.2}$$

Where $\Lambda_s = \int_0^t \beta_t\, ds$, and $\Lambda_t - \Lambda_s = \int_s^t \beta_s\, ds$ denote the decay factor. It can be observed from Equation (4.2), as $t \to \infty$, the solution for X_t converges toward the stationary distribution, $X_t \sim \mathcal{N}(\mu, \Sigma)$, where $\mathcal{N}(\mu, \Sigma)$ is invariant for all t.

The application of the the SDE as the forward process allows sampling from the initial empirical distribution $X_0 \sim p_{\text{data}}(x)$, and provides analytical solutions to the noising process, as the distribution tends to noise. This setup enables interpretation of the forward process as a reference dynamic that smooths or degrades data samples, similar to diffusion models (Ho et al. 2020).

In SGMs, the noise-conditional score functions corresponding to the data density, perturbed by the forward SDE, are linked via the Fokker-Planck (FP) equation. The FP equation describes how the probability distribution of a variable changes over time when the variable is subject to random forces and drift applied by the OU process (Lai et al. 2023). Through application of the FP equation, the probability density function $p_t(x)$ can be expressed for any time t by,

$$\partial_t\, p_t(x,t) \;=\; \tfrac{\beta_t}{2}\,\nabla_x \cdot \big(\Sigma^{-1}(x-\mu)\, p_t(x)\big) \;+\; \tfrac{\beta_t}{2}\,\Delta_x p_t(x), \tag{4.3}$$

By considering a probability density that is described by the Boltzmann distribution, $p_t(x) \propto \exp(-E(x,t))$, and perturbed by the FP equation, the density $p_t(x)$ is evolved from $p_{\text{data}}(x)$ toward the stationary distribution as $\mathcal{N}(\mu, \Sigma)$ as $t \to \infty$. Note that the partition function Z_t is not needed when computing gradients or derivatives with respect to x, and thus cancels out in the FP dynamics.

To characterise the transport of probability mass in the FP equation, a transition kernel for the process can be described by the Gaussian,

$$p(x_t \mid x_0) = \mathcal{N}\left(x_t \,\middle|\, x_0\, e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} + \mu(1 - e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s}),\ \frac{\sigma^2}{2\theta}\left(1 - e^{-\Sigma^{-1}\Lambda_s}\right) I\right) \tag{4.4}$$

The kernel provides the conditional transition density required for simulating trajectories of the density. It serves as the Green's function of the Fokker–Planck equation, capturing the probability of transitioning from x_0 to x_t over time t under the forward dynamics.

To simulate sample paths from the forward process, the analytical closed form discretisation is applied to the SDE. For a fixed time step $\Delta\Lambda_s$, resulting in the update equation,

$$X_{t+1} = X_t e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s} + (1 - e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s})\mu + \sqrt{\sigma^2(1 - e^{-\Sigma^{-1}\Delta\Lambda_s})} \cdot Z_t \tag{4.5}$$

where $Z_t \sim \mathcal{N}(0, I)$ are independent Gaussian noise terms. This update scheme corresponds to the exact transition for each X_t over discrete time steps $\Delta\Lambda_s$. By iterating these updates, a discrete trajectory $\Lambda_s = t \cdot \Delta\Lambda_s$ obtains an approximation for the continuous-time dynamics while preserving the correct mean and covariance at each step.

This forward process defines the time-indexed distribution $p_t(x)$ and transition kernel required for constructing Feynman–Kac expectations, which in principle yield the reverse-time energy functional. In practice, however, we employ Langevin dynamics for tractable sampling, which approximates these marginals rather than preserving them exactly.

4.3 Reverse Process

Feynman-Kac Application

The forward process describing the time evolution of a probability density $p_t(x)$, under stochastic diffusion and drift, is governed by the Fokker-Plank formula shown in Equation (4.3). This equation can be expressed as a deterministic PDE involving the infinitesimal generator $\mathcal{L}_x^*$,

$$\frac{\partial p(x,t)}{\partial t} = \mathcal{L}_x^*\, p(x,t) \tag{4.6}$$

where $\mathcal{L}_x^*$ is the adjoint of the infinitesimal generator of the underlying SDE, encoding the drift and diffusion contributions to the density evolution, with the initial condition, $p_0(x) = p_{data}(x)$. From this formulation, the density evolution can be traced forward through time via a transition kernel, shown by Equation 4.4. Crucially, there exists a corresponding reverse

process, through which, the PDE is back-propagated from the terminal distribution $p_T(x)$, thereby recovering $p_{data}(x)$. This time inversion aligns with the formulation provided by the Feynman-Kac equation.

The Feynman-Kac (FK) formula states the connection between linear parabolic PDEs, and expectation of stochastic processes driven by Brownian motion (Pham 2014). Intuitively, the FK theorem asserts that over sufficiently many stochastic realisations, the empiric distribution of sampled paths converges to the solution of the associated PDE, effectively representing the transition from the terminal distribution back to the initial distribution, $p(x,T) \to p_{data}(x)$. If allowed exhaustive partitioning over the domain and infinitely many trajectories simulated, the solution of the FK reconstructs the initial distribution. This is intractable in practice and due to stochasticity induced in the forward process, and limited number of path realisations simulated, an approximation of the initial distribution is reconstructed, $p_{data}(x) \sim \hat{p}_{data}(x)$. This estimator underlies the construction of the energy functional derived in section 4.4, which in turn defines the generative process via Langevin dynamics.

4.4 Feynman-Kac Process

In this section we derive an analytical construction of the reverse process under the FK formulation by using an non-parametric energy scalar, $E(x,t)$. We also provide a intuitive description of the free energy, linking the gradient flow, shown in Equation 2.15 with the variational Bayesian notion of free energy. Both describe a functional combining expected energy and entropy, whose minimisation governs system dynamics. Firstly, consider a general linear parabolic PDE of form,

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) + \mathcal{L}_x u(x,t) = V(x,t)u(x,t) + g(x,t) \\ u(x,T) = f(x) \end{cases} \tag{4.7}$$

Where $u(x,t)$ denotes the solution to the FK, which can be interpreted as a time-dependent partition function, and f is a bounded measurable function. If $V(x,t)$ is a continuous function of $(x,t) \in [0,\infty) \times \mathbb{R}^d$, then there exists general FK formula (Del Moral 2004),

$$u(x,t) = \mathbb{E}_{X_s \sim Q}\left[f(X_T) \exp\left(-\int_t^T V(X_s,s)\, ds \right) + \int_t^T g(X_\tau,\tau) \exp\left(-\int_t^\tau V(X_r,r)\, dr \right) d\tau \right] \tag{4.8}$$

Where X_T is the solution to a forward Itô diffusion in $\mathbb{R}^d$ with generator $\mathcal{L}_x$ defined by the OU drift and diffusion contributions,

$$\mathcal{L}f = \theta(\mu - X_t) \cdot (\nabla f)^T + tr\left((\nabla^2 f) \cdot \frac{\sigma^T \sigma}{2} \right)$$

Functions $f(X_t)$ and $g(X_\tau,\tau)$ define the terminal and source contributions to the solution. The terminal function $f(X_t)$ prescribes the final value $u(x,T)$, and serves as a boundary condition that determines the endpoint constraint of the stochastic process. The terminal

function acts essentially as the final cost or bias, influencing the expected outcome of the backward-evolving solution $u(x,t)$. In contrast, the source term $g(X_\tau, \tau)$ models the distributed forcing along the trajectory. Where it introduces time-dependent adjustments to the energy landscape, capturing local perturbations or external outputs that accumulate over time and effect the solution path-wise, acting as a form of regularisation.

To interpret the role of the scalar energy $E(x,t)$, it is useful to first connect the free energy concepts introduced in by Equation 2.17 in Preliminaries section 2.2. As demonstrated, the statistical mechanics notion of the Helmholtz free energy and the variational Bayes formulation reduce to a functional combining expected energy and entropy, whose minimisation induces gradient flows such as the OU and FP dynamics. The Free Energy Principle (FEP) (Friston 2010) extends the same idea to cognitive systems, framing dynamics as the minimisation of the variational free energy. This concept provides a variational Bayesian interpretation of the non-parametric energy function, giving analogy to how the system self-organises, aiding to clarify how the scalar energy $E(x,t)$ induces the vector field dynamics through its gradient $-\nabla_x E(x,t)$, describing the mass transportation during the generative process.

Free Energy Interpretation The free energy principle (FEP) states that for any self-organising system at equilibrium with its environment, it must naturally minimise its free energy, effectively resisting a natural tendency to disorder. In biological systems this manifests physiologically as the maintenance of homeostasis (Friston 2010). Analogously, in the context of a generative model, the distribution $\hat{p}_{data}(x)$ can be viewed as a phenotype; an emergent representation, shaped via interaction with its underlying environment. Under the variational Bayes perspective, minimisation of free energy biases the system toward low-surprisal (high-probability) states, which correspond to regions of relatively low entropy and can be interpreted as homeostatic equilibria.

From variational Bayesian inference the free energy can be seen as an upper bound on the surprisal of particular states, compromising of the complexity and accuracy terms (Bettinger & Friston 2023). Translated, this corresponds to the difference in the expected energy and the entropy (Friston et al. 2023),

$$\begin{aligned}\mathcal{F}[\pi(\tau)] = \underbrace{\mathbb{E}_q[-\log p(x, \eta(\tau), \pi(\tau))]}_{\text{Energy expectation / accuracy term}} + \underbrace{\mathbb{E}_q[\log q(\eta(\tau))]}_{\text{Negative entropy / complexity term}} \\ = \mathrm{KL}(q(\eta(\tau)) \,\|\, p(\eta(\tau) \mid x, \pi(\tau))) \; - \; \log p(x \mid \pi(\tau)).\end{aligned} \tag{4.9}$$

This equivalence shows that free energy minimisation is identical to maximising the evidence lower bound (ELBO), since the first term is the KL divergence between the approximate posterior $q(\eta(\tau))$ and the true posterior $p(\eta(\tau) \mid x, \pi(\tau))$, up to the constant $-\log p(x \mid \pi(\tau))$. Here, $\pi(\tau)$ denotes a control policy over time, $\eta(\tau)$ are the latent trajectories (corresponding to X_s in the FK path integral, Equation 4.8), and $q(\eta(\tau))$ is the approximate posterior over the path distribution $Q[X_s]$. The expected energy is given by the negative log joint probability, $E(x) = -\log p(x, \eta(\tau), \pi(\tau))$, while the entropy term corresponds to the negative differential entropy of $q(\eta(\tau))$. By simplifying this expression, and applying the Helmholtz free energy

definition, the variational free energy takes the form of the negative log of a time-dependent partition function,

$$\mathcal{F}[q] = \mathbb{E}_q[E(x)] - H[\eta(q)] = -\log Z \tag{4.10}$$

In the FP formalism, the steady state density, $p(x)$, satisfies a stationary solution where $\partial_t p = 0$. From this relationship, under assumptions of a gradient flow ($-\nabla E(x)$), the energy of the state system relative to energies of all possible states forms a Boltzmann distribution Hinton (2017),

$$p(x) \propto e^{-E(x)}$$

Thus, the energy function $E(x)$ can be recovered as the negative log of the steady state density, up to a constant. As $t \to \infty$, the FK solution converges asymptotically toward the stationary FP solution, thus the time inhomogeneous energy $E(x,t) = -\log u(x,t)$, recovers the steady state energy $E(x)$ (Wang 2025),

$$E(x) = \lim_{t\to\infty} E(x,t) \qquad p(x) = \lim_{t\to\infty} u(x,t) \tag{4.11}$$

Here, $u(x,t)$ plays the role of a *time-dependent partition function*, obtained directly through the FK path expectation. It generalises the constant Z of the Boltzmann distribution, and its negative log defines the inhomogeneous energy,

$$E(x,t) := -\log u(x,t) \tag{4.12}$$

By applying this definition to Equation 4.8, the scalar energy expression is found,

$$E(x,t) := -\log \mathbb{E}_{X_s\sim Q}\left[f(X_T)\exp\left(-\int_t^T V(X_s,s)\,ds\right) + \int_t^T g(X_\tau,\tau)\exp\left(-\int_t^\tau V(X_r,r)\,dr\right)d\tau\right] \tag{4.13}$$

4.4.1 Discretisation of the Forward Process via Brownian Motion

To simulate sample paths from the forward process, the time interval is partitioned into K uniform steps,

$$t^{(k)} = s_0 < s_1 < \cdots < s_K = T, \quad \Delta s = s_{j+1} - s_j$$

We generate M independent trajectories in parallel. Each trajectory $i \in 1,\ldots,M$ is initialised at a data point $X_{s_j}^{(i)}$, chosen from a mini-batch of samples from the source distribution X_0. Thus, each trajectory corresponds to one Monte Carlo sample path starting at its own initial condition. From these initial states, the trajectories evolve independently according to the stochastic differential equation in Equation 4.1. In practice, we apply the standard discretisation scheme for Brownian motion to obtain successive states $X_{s_j}^{(i)}$ for $j = 1,\ldots,K$,

$$X^{(i)}_{s_{j+1}} = X^{(i)}_{s_j} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} + (1 - e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s})\mu + \sqrt{\sigma^2(1 - e^{-\Sigma^{-1}\Lambda_s})} \cdot Z^{(i)}_j$$

where $Z_j \sim \mathcal{N}(0, I)$ are i.i.d. standard Gaussian noise vectors. This defines a discrete approximation of the continuous Brownian trajectory with drift, starting from sample $x^{(k)}$. The final distribution informs the reverse process. The forward process can be applied through the following algorithm,

Algorithm 1 Simulation of the Forward Process via Ornstein-Uhlenbeck Process

Require: Number of trajectories M, number of steps K, time increment $\Delta\Lambda_s$, mean μ, covariance Σ, initial batch $\{x^{(i)}_0\}_{i=1}^M$

Ensure: Discrete trajectories $\{X^{(i)}_t\}$ for $i = 1, \ldots, M$, $t = 0, \ldots, K$

1: **for** $i = 1$ to M **do** ▷ Loop over each trajectory
2: Initialise $X^{(i)}_0 \leftarrow x^{(i)}_0$
3: **for** $t = 0$ to $K - 1$ **do** ▷ Loop over time steps
4: Sample $Z^{(i)}_t \sim \mathcal{N}(0, I_d)$
5: Update trajectory using OU transition:

$$X^{(i)}_{t+1} = X^{(i)}_t \, e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s} + \big(I - e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s}\big)\mu + \sqrt{\sigma^2\big(1 - e^{-\Sigma^{-1}\Delta\Lambda_s}\big)}\, Z^{(i)}_t$$

6: **end for**
7: **end for**

4.4.2 Feynman-Kac Reverse Process Application

Under the FK formulation, by finding the derivative expression of the scalar energy $\nabla_x E(x, t)$ we derive a closed-form expression for the reverse sampling process. Recalling Equation 4.13, the time inhomogeneous energy is defined by,

$$E(x, t) = -\log \mathbb{E}_{X_s \sim Q}\left[f(X_T) \exp\left(-\int_t^T V(X_s, s)\, ds \right) + \int_t^T g(X_\tau, \tau) \exp\left(-\int_t^\tau V(X_r, r)\, dr \right) d\tau \right]$$

Allowing source time $\tau = s_j$, and discretising over K time steps, the expression can be simplified into two discrete components, the terminal β and source γ components,

$$\gamma^{(i)} = f(X^{(i)}_T) \exp\left(-\sum_{j=0}^{K-1} V(X^{(i)}_{s_j}, s_j)\, \Delta s \right), \tag{4.14}$$

$$\beta^{(i)} = \sum_{j=0}^{K-1} g(X^{(i)}_{s_j}, s_j) \exp\left(-\sum_{r=0}^{j-1} V(X^{(i)}_{s_r}, s_r)\, \Delta s \right) \Delta s. \tag{4.15}$$

This gives the simplified expression for the scalar energy function as,

$$E(x,t) = -\log \mathbb{E}_{X_s \sim Q}\left[\gamma^{(i)} + \beta^{(i)} \middle| X_t = x\right]$$

To approximate the expectation of $E(x,t)$, we employ Monte Carlo methods. Since the gradient of this scalar defines the probability flow, we interpret it as a *potential*. Its Monte Carlo approximation will henceforth be referred to as the *Monte Carlo Potential Estimator (MCPE)*. Application of MC methods for sampling multiple path trajectories allows for efficient analysis of the defined energy system. The use of MC is supported for simulating Boltzmann statistics, so seems a natural fit for application to the FK problem. Averaging the multivariate integral over multiple paths offers the mean trajectory tending toward equilibrium, the full MCPE expression is as follows,

$$\hat{E}(x,t) = -\log\left[\frac{1}{M}\sum_{i=1}^{M}\left(\gamma^{(i)} + \beta^{(i)}\right)\right] \tag{4.16}$$

This application of the Monte Carlo estimator allows for efficient computation, by reducing the multidimensional path integral to a simple average over sampled trajectories, as well as providing statistical convergence, given sufficient path sampling. By re-expressing the aggregate of the $\gamma^{(i)}$ and $\beta^{(i)}$, the MCPE is further refined,

$$\alpha^{(i)} = \gamma^{(i)} + \beta^{(i)} \tag{4.17}$$

Allowing the simplified re-expression to take the form,

$$\hat{E}(x^{(i)}, t^{(i)}) = -\log\left[\frac{1}{M}\sum_{i=1}^{M}\alpha^{(i)}\right] \tag{4.18}$$

Under this simplification, the Monte Carlo partition function is expressed as the average sum of the $\alpha^{(i)}$ terms over each M path,

$$\hat{Z}(x,t) = \frac{1}{M}\sum_{i=1}^{M}\alpha^{(i)} \quad \Longrightarrow \quad \hat{E}(x,t) = -\log \hat{Z}(x,t) \tag{4.19}$$

To define the vector field, the gradient of the MCPE must be have closed-form expressions to remain interpretable. This can be found through the derivative expression,

$$\nabla_x \hat{E}(x,t) = -\sum_{i=1}^{M}\hat{\alpha}^{(i)}\nabla_x \log \alpha^{(i)} \tag{4.20}$$

where the normalised weights are given by,

$$\hat{\alpha}^{(i)} = \frac{\alpha^{(i)}}{\sum_{i=1}^{M} \alpha^{(i)}} \tag{4.21}$$

If functions $V(X_{s_j}, s_j)$, $f(X_T)$ and $g(X_{s_j}, s_j)$ have differentiable analytical solutions, the expression retains a fully closed-form solution. The final full expression for the MCPE gradient (GradMCPE) when expanded to include the terminal, source and potential function has the general form (see Appendix A.1.5 for full derivation),

$$\nabla_x \hat{E}(x^{(i)}, t^{(i)}) = \sum_{i=1}^{M} \hat{\alpha}^{(i)} \left[- \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)})\, e^{-\theta_{s_j}}\, \Delta s + \psi^{(i)} \right] \tag{4.22}$$

where $e^{-\theta_{s_j}}$, represents the attenuation factors from the diffusion process and $\psi^{(i)}$ captures the contributions from the terminal and source function gradients,

$$\psi^{(i)} := \frac{\nabla_{X_T^{(i)}} f(X_T^{(i)})\, e^{-\theta_T} + \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} g(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s}{f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\, \Delta s}$$

The GradMCPE can be implemented using the following algorithm, that when implemented induces the vector field with controllable bias and regularisation through the terminal $f(X_T)$ and source $g(X_{s_j}, s_j)$ respectively.

Algorithm 2 GradMCPE: Gradient of the Monte Carlo Potential Estimator

Require: Number of trajectories M, number of steps K, step size Δs, functions $V(x,t)$, $f(x)$, $g(x,t)$, trajectories $\{X_{s_j}^{(i)}\}$, forward noise schedule $\{\Lambda_{s_j}\}_{j=1}^{K}$, with decay map $\Theta_{s_j} = \frac{1}{2}\Sigma^{-1}\Lambda_{s_j}$

Ensure: Gradient of the Monte Carlo Potential Estimator, $\nabla_x \hat{E}(x,t)$

Compute reverse attenuation factors from the forward decay:

$$\text{for each } j = 1, \ldots, K, \quad \theta_{s_j} \leftarrow e^{-\Theta_{s_j}}$$

Initialise terminal factor $\theta_T \leftarrow e^{-\Theta_T}$

Initialise source factors $\{\theta_{s_j}\}_{j=0}^{K-1}$ accordingly

for $i = 1$ to M **do**

Compute terminal component:

$$\gamma^{(i)} \leftarrow f(X_T^{(i)}) \exp\left(- \sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\, \Delta s \right)$$

Compute source component:

$$\beta^{(i)} \leftarrow \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\, \exp\left(- \sum_{r=0}^{j-1} V(X_{s_r}^{(i)}, s_r)\, \Delta s \right) \Delta s$$

Aggregate:

$$\alpha^{(i)} \leftarrow \gamma^{(i)} + \beta^{(i)}$$

end for

Normalise weights:

$$\hat{\alpha}^{(i)} \leftarrow \frac{\alpha^{(i)}}{\sum_{m=1}^{M} \alpha^{(m)}}$$

for $i = 1$ to M **do**

Compute gradient contributions from V, f, and g:

$$\psi^{(i)} \leftarrow \frac{\nabla_{X_T^{(i)}} f(X_T^{(i)})\, e^{-\theta_T} + \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} g(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s}{f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\, \Delta s}$$

Compute potential gradient term:

$$G^{(i)} \leftarrow - \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s \; + \; \psi^{(i)}$$

end for

Aggregate final gradient:

$$\nabla_x \hat{E}(x,t) \leftarrow \sum_{i=1}^{M} \hat{\alpha}^{(i)}\, G^{(i)}$$

4.4.3 Sampling Procedure

Langevin Sampling

By defining the derivative energy flow show in Equation 4.28, samples can be drawn from the distribution using Langevin dynamics since $-\nabla_x E(x^{(i)}, t^{(i)}) = \nabla_x \log p_t(x)$,

$$x_{t+1} = x_t - \frac{\epsilon}{2}\nabla_x \, E(x^{(i)}, t^{(i)}) + \sqrt{\epsilon}\, z_t \quad z_t \sim \mathcal{N}(0, 1) \tag{4.23}$$

Where the process at x_0 is initialised from the final distribution from the forward noising process under the OU scheme seen in Equation 4.5. While Langevin dynamics is employed here for tractability and simplicity, the framework is not restricted to this choice, alternative sampling schemes such as probability flow ODEs or Hamiltonian methods could also be applied. Our choice of Langevin dynamics reflects its direct compatibility with the potential formulation and its efficiency for Monte Carlo estimation.

4.5 Kernel Descriptions

Kernel estimates are employed in the proposed framework to represent the potential $V(X_s, s)$, terminal $f(X_T)$, and source $g(X_s, s)$. We adopt kernels because they are non-parametric, interpretable, and computationally efficient under Monte Carlo methods. Their bandwidth (kernel width) directly controls locality versus global smoothness, offering an intuitive mechanism to regulate sharpness of the induced energy landscape. Moreover, kernels are distance-based and readily extendable with weighting schemes, enabling explicit control over contribution of different samples. In this study, we employ normalised Gaussian radial basis function (RBF) kernels, which provide a flexible and tractable density representation compatible with the FK formulation.

RBFs describe a real valued function φ, whose value depends on the input x_i, and target data x, so that $\varphi = \hat{\varphi}(||x - x_i||)$ decreases monotonically in value as the query and sample converge (Orr et al. 1996). For a normalised Gaussian RBF with temperature scaling, the kernel and its derivative is defined by,

$$\varphi(x, x_i) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \qquad \nabla_x \varphi(x, x_i) = -\frac{x - x_i}{\sigma^2}\, \varphi(x, x_i) \tag{4.24}$$

By normalising the function, φ can be treated as a density function since $\int_{\mathbb{R}^d} \varphi(x, x_i)\, dx = 1$, this moves away from pointwise features to a distribution over features $\varphi(x, x_i) = p(x)$, moving learning to the probability space. This normalisation makes the hidden units perform a function similar to that of a *soft* Voronoi tessellation over the input space, allowing overlap between regions, that when un-normalised would sharply partition (Bugmann 1998). For an un-normalised RBF, the kernel's activations can be dominated by a singular large response, causing mode collapse into a singular prototype. Normalisation blurs the boundary by expressing each input as a distribution over all kernels, mitigating numerical instability and mode collapse, whilst promoting generalisation through smoother node transition. Applying normalisation over all input space promotes global smoothing. Overly diffuse weighting

therefore forms highly flat energy space, to control the localisation of the soft partitions, a computationally inexpensive temperature scaling of the kernel values is applied that obeys thermodynamic regulation,

$$w^{(i)}(x) = \frac{\exp\left(\frac{\log \varphi(x,x_i)}{\kappa_i}\right)}{\sum_j \exp\left(\frac{\log \varphi(x,x_j)}{\kappa_j}\right)} \tag{4.25}$$

where κ_i denotes a dimensionless parameter enforcing the temperature curriculum. For values where $\kappa_i < 1$, the contrast between kernels increases, sharpening the partitions, hence increasing locality and forming sharper *valleys* in the energy landscape, causing more probable, low energy samples to dominate. If $\kappa_i = 1$, the normalised RBF equates to a standard normalised KDE, providing a smooth distribution, where each kernel contributes proportionally. In the case where $\kappa_i > 1$, the differences in $\log \varphi$ become increasingly compressed, as $\kappa \to \infty$ weights become increasingly uniform across kernels, forming a highly flat energy landscape, reducing local structure and increased global smoothing.

Through promotion of the RBF into log-sum-exponential (LSE) (or RealSoftMax) space we redefine the kernels such that weighting emerges naturally in the derivatives, allowing explicit shaping of the score field through κ_i. This redefinition under the LSE forms a smooth approximation of the maximum function, $\mathrm{LSE}(x_1, \ldots, x_n) = \log(e^{x_1} + \cdots + e^{x_n})$, this application reduces underflow and overflow issues, increasing accuracy of the MCPE and promoting stability,

$$\log p(x) = \mathrm{LSE}(\varphi) = \log \sum_{i=1}^{N} e^{\log \varphi(x,x_i)/\kappa_i} \tag{4.26}$$

where the partial derivative w.r.t. x forms the softmax weights $w^{(i)}$ implicitly,

$$\nabla_x \log p(x) = \sum_i \underbrace{\frac{e^{\log\ \varphi(x,x_i)/\kappa_i}}{\sum_j e^{\log\ \varphi(x,x_j)/\kappa_j}}}_{\text{softmax weights, } w^{(i)}(x)} \nabla_x \varphi(x, x_i) \tag{4.27}$$

Since the overall energy function $E(x,t) = \log p(x)$ is defined from Boltzmann form 2.2, $p(x) \propto e^{-E(x)}$, consistent with the EBM formulation provided in 4.12, the kernel weights $w^{(i)}$, are computed via a temperature scaled softmax over $\log \varphi(x, x_i)$. This scaling therefore modulates the the localisation of the energies influence, acting as a controllable regulisers in the CFFM framework to influence smoothness of the induced energy landscape, whilst retaining the Boltzmann-defined probabilistic structure at the energy level.

The final derivative energy function in terms of the LSE defined RBF kernels, under the MCPE gradient found in Equation 4.22 is as follows,

$$\nabla_x E(x^{(i)}, t^{(i)}) = \sum_{i=1}^{M} \hat{\alpha}^{(i)} \left[\sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} \varphi(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s \; + \psi^{(i)} \right] \tag{4.28}$$

where,

$$\psi^{(i)} = \frac{\nabla_{X_T} \varphi(X_T^{(i)})\, e^{-\theta_T} \; + \; \sum_{j=0}^{K-1} \nabla_{X_{s_j}} \lambda_{s_j}^{(i)}\, \varphi(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s}{\varphi(X_T^{(i)}) + \sum_{j=0}^{K-1} \; \lambda_s^{(i)} \varphi(X_{s_j}^{(i)}, s_j)\, \Delta s}$$

By defining the kernels in LSE space, the construction interpolates between a standard kernel density estimate when the temperature parameter yields uniform weighting, and a competitive softmax-weighted density when the temperature sharpens. The LSE RBF therefore formulates a *competitive KDE*, where kernels smoothly compete to explain each query point, allowing explicit control over locality versus global smoothing through κ_j.

Chapter 5

Results

5.1 Validation Tests on Toy Data

To demonstrate the generation ability using the CFFM, we initially apply the model under different temperature schemes to demonstrate the impact on kernel locality. For comparisons, we apply the model to various toy datasets in order to demonstrate the general behaviour of the model to differently shaped data.

In application of the model, the terminal, source and potential functions convey different purpose. To naïvely differentiate effects between kernels, direct fractional scaling to the temperature is applied. For the potential $V(X_{s_j}, s_j)$, the temperature scaling applied initially is held at a constant value, κ. The potential value is the unbiased flow, that encodes the broad geometry of the data distribution, shaped by the the terminal and source functions, providing uniformity to weighting for initial demonstration provides a clearer effect of the explicit bias. The terminal function ascribes the final condition of the energy $E(x, T)$ or the *goal constraint* that represents the target distribution, specifically it acts as the bias toward the data manifold at final time $t = T$. For the initial results, the terminal distribution enforces concentration around modes of the data. The source functions role is to model continuous injection or removal of conserved quantities, such as mass or heat for example, at intermediate times. In the CFFM framework, it appears as the path-integrated contribution of the energy at time t, biasing trajectories according to how much they accumulate or dissipate along each trajectory. Essentially, the source function should encourage exploration over the manifold, implying its temperature weighting encourage uniformity. To differentiate the relative effects of each kernel in the naïve demonstration, the potential kernel is assigned $\tau_V = \kappa$, the source kernel is given a greater uniformity weighting $\tau_G = 1.2\kappa$, and the terminal kernel a sharper weighting $\tau_F = 0.1\kappa$. These relative values are arbitrary and selected only to illustrate how fractional rescaling of the temperature parameter alters the bias of each kernel component. To first show effect of the baseline κ on manifold convergence, we apply the model to the same toy data.

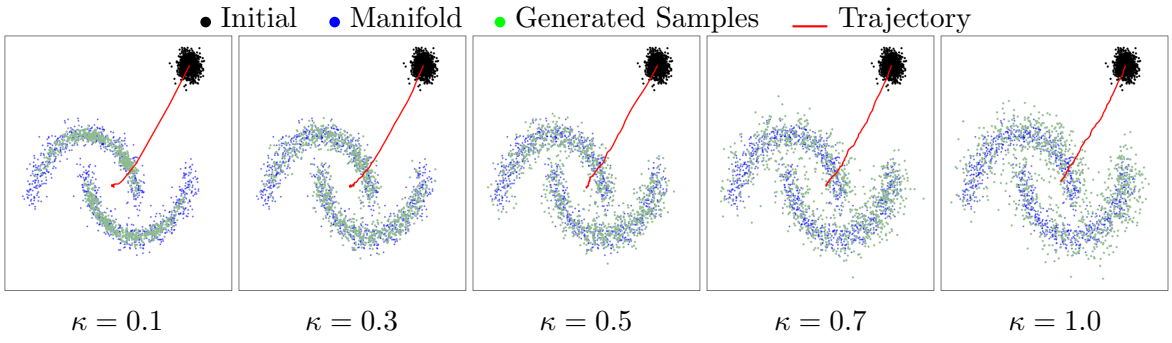


Figure 5.1: Many Moons generated samples. Demonstration of generated data of the Gaussian kernel at different temperature values, κ. The average trajectory tracks convergence of intermediate samples from the initial distribution to the manifold from $t = 0$ to $T = 200$.

Figure 5.1 demonstrates the impact of temperature on manifold convergence to non-linear manifolds. It is observed that as the temperature increases, the dissipation of generated points over the manifold increases. Lower temperatures observe tight collapse onto the target distribution, where a majority of the mass is centred at segments of the manifold closest to the distribution mean. As the temperature is increased, the generated data still captures the shape of the distribution, where higher temperatures show a broader coverage of the manifold. However, though the model still captures the shape of the distribution, generated data starts to fall off the manifold.

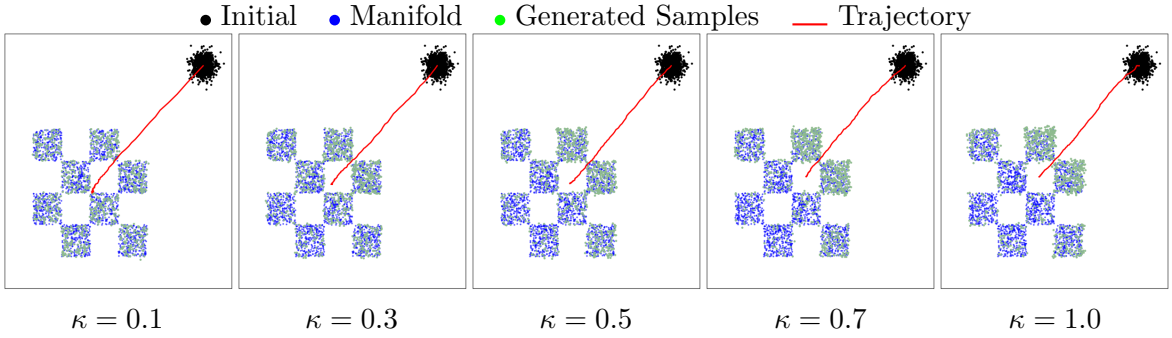


Figure 5.2: Chequerboard generated samples. Demonstration of generated data of the Gaussian kernel at different temperature values, κ. The average trajectory tracks convergence of intermediate samples from the initial distribution to the manifold from $t = 0$ to $T = 200$.

Figure 5.2 applies the model to an alternative dataset that represent more periodic structure. At lower temperature, the weighting in the kernel is sharper, particles are strongly pulled toward the manifold, providing better mode coverage. The average trajectory penetrates more deeply, aligning with the manifold structure more. Overall mass coverage seems equally distributed among nodes. At higher temperatures, the figure shows the effect of smoother kernels, implying more uniform weighting. Mass tends to group at the closest node to the

initial distribution, indicating the model is distance invariant. The average trajectory shows shallower penetration and limited exploration. Nonetheless, the model is able to identify the manifold, and under periodic structure it appears more clearly at higher temperatures. In this regime, probability mass remains on the manifold, leading to a reduction in artefacts. However, the overall effect of higher temperature with the same number of sampling steps is a tendency toward underfitting given limited sampling steps.

Across both datasets (Figures 5.1 and 5.2), temperature modulates how kernels discriminate by distance. At lower temperatures, kernels are sharper, trajectories penetrate more deeply, producing tight alignment with manifold structure (clearer in the chequerboard case). As temperature increases, kernels become smoother, weights more uniform, and generated mass tends to cluster near the initial distribution rather than tracking the manifold geometry. This behaviour highlights temperature as a trade-off between exploration and concentration. At lower temperatures, kernels drive sharper convergence but also risk collapse, with samples remaining on the manifold but concentrating near the mean due to the OU process's mean-reversion dynamics. At higher temperatures, kernels smooth out, producing broader coverage across the manifold but reducing local fidelity. These findings suggest that a temperature annealing schedule, beginning high to encourage exploration, then reducing to sharpen convergence, may strike a balance between coverage and stability.

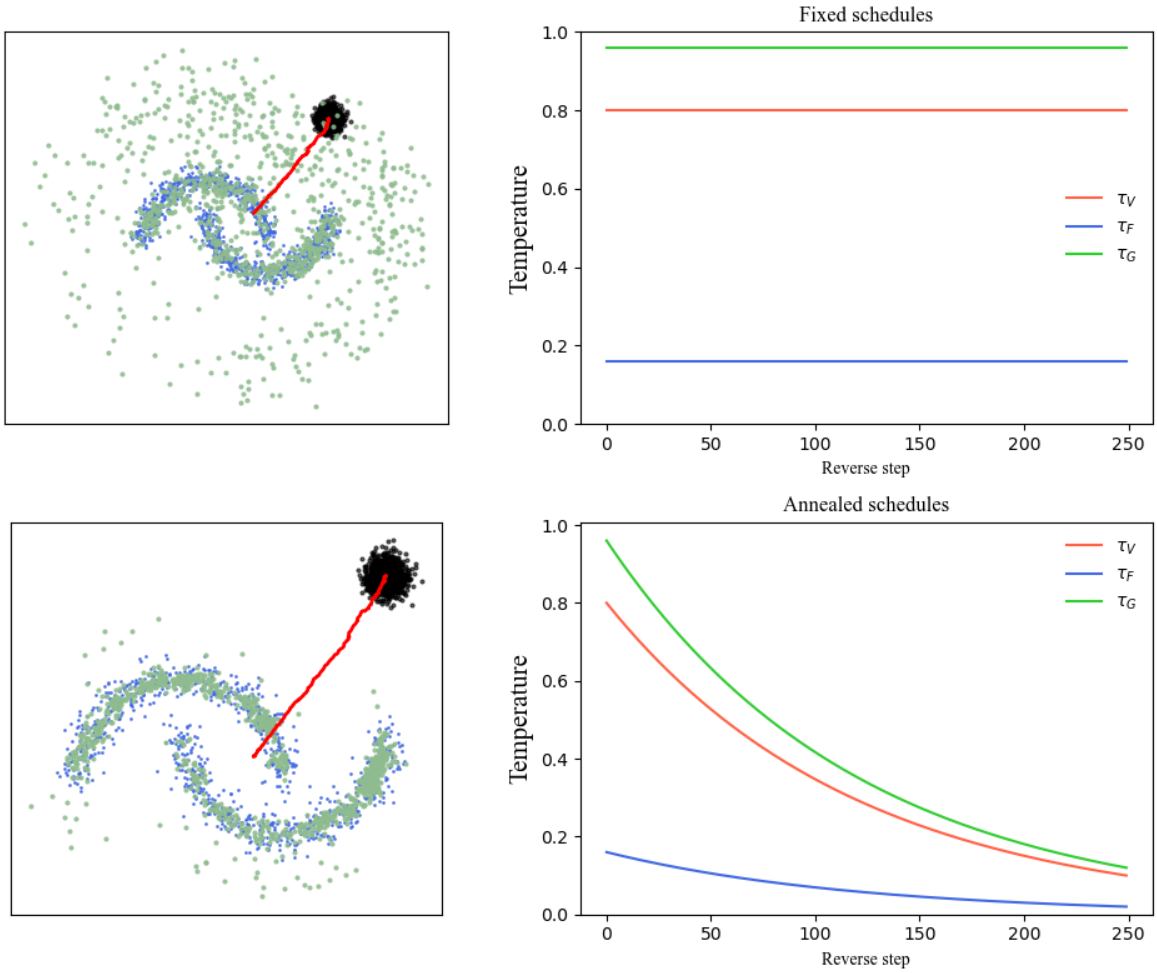


Figure 5.3: Annealed Temperatures. Demonstration of generated data of the Gaussian kernel for fixed temperatures and annealed temperatures. The average trajectory tracks convergence of intermediate samples from the initial distribution to the manifold from $t = 0$ to $T = 200$, using fixed KDE bandwidth and step size.

To demonstrate the effect of temperature annealing, Figure 5.3 demonstrates a fixed temperature compared to the annealed temperature regime. Whilst fixed temperatures do show convergence, many artefacts appear in the final generated distribution. In comparison, under the annealed temperature, the generated data contains less artefacts, with convergence spreading more uniformly over the manifold. By using temperature annealing, the kernels gain the attributes of broader manifold coverage at higher temperatures during early sampling, with the convergence strength at lower temperatures, combining attributes observed in Figure 5.1 to invoke better distribution coverage. To demonstrate the full reverse process under the temperature annealing, we simulate the evolution using KDE plots to show the transition of the distribution density.

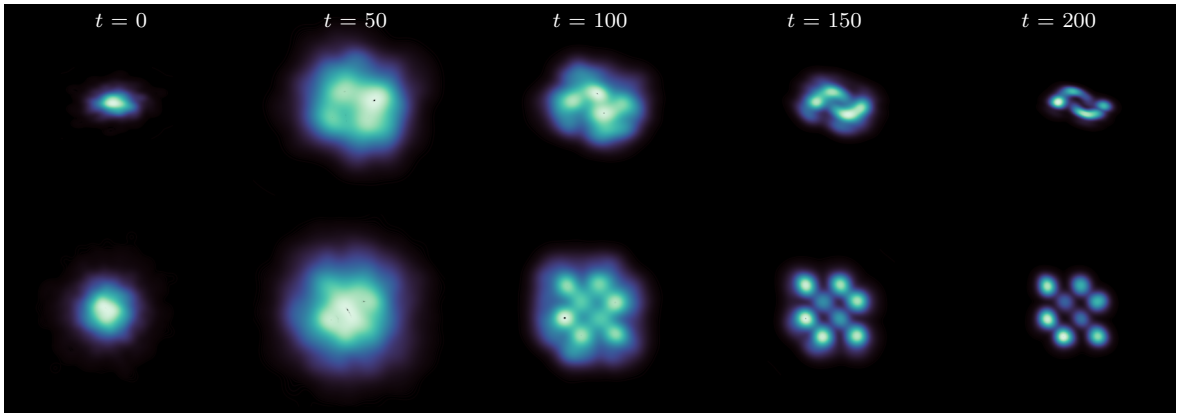


Figure 5.4: Manifold recovery from forward noising process. Demonstration of manifold recovery from the forward noising process, tracked through KDE plots to show mass groupings over the reverse process using temperature annealing. Top row shows the many moons toy data reconstruction, bottom row shows the chequerboard evolution, plots were formed with fixed step size ϵ, and KDE bandwidth σ.

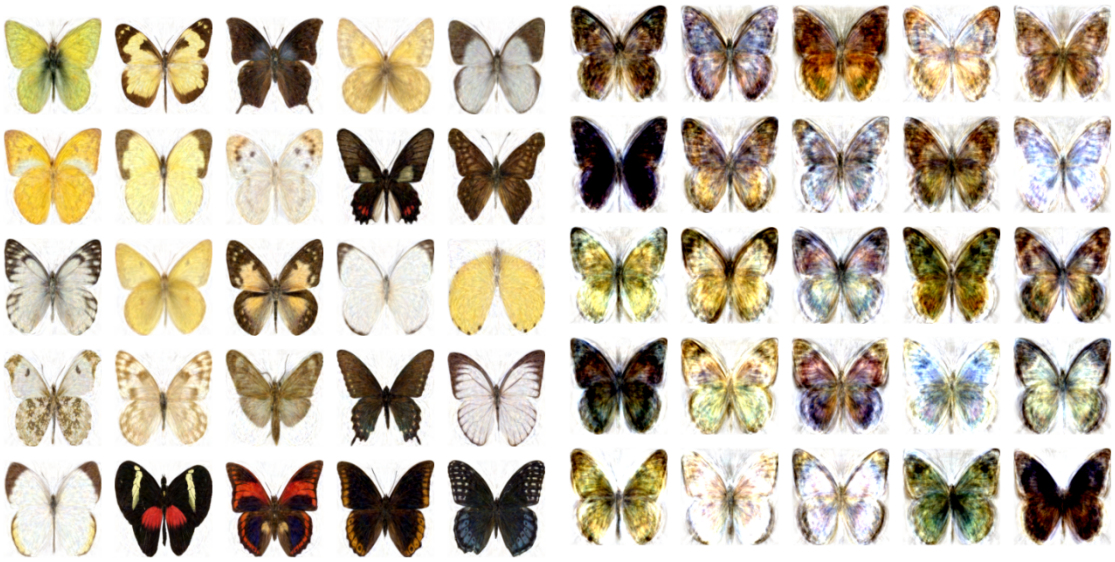
Figure 5.4 illustrates the denoising process, where the initial distribution at $t = 0$ corresponds to the fully diffused toy data produced by the forward process. As the reverse simulation proceeds, the distribution progressively contracts back toward the original data manifold. In the early stages ($t \approx 50$), the distribution appears greatly expanded relative to the data, a consequence of the forward process injecting high variance. The reverse dynamics counteract this by applying strong smoothing at higher temperatures, which initially exaggerates dispersion but facilitates exploration across the manifold before convergence. The effective result of this smoothing is that in later stages, the model captures the full distribution, converging to the manifold. By the final time step, $t = 200$, the underlying manifold is largely recovered. However, the reconstructed density exhibits concentration at the extremities of the manifold. In the many-moons case, mass accumulates first at the ends of the crescents, followed by a brief low-density gap before the distribution gradually smears along the full crescent structure. In contrast, the periodically structured chequerboard domain represents a piecewise-linear distribution - allocating more density to the extremity nodes - but overall shows a more uniform mass convergence, with density directly partitioning into node-like cells.

The behaviour in the many moons case may be attributed to the non-linearity in the manifold. Neural generative models implicitly parametrise non-linear data manifolds, the curvature of these manifolds influences the geodesics and density propagation (Ma & Fu 2011). Whilst to our knowledge no prior work explains the observed phenomena of the 'tailing' or gap formation along the many-moons manifold, the resultant density along the manifold echoes similar behaviours in Riemannian flow-based models, where regions of higher curvature or complex topology act as focal points during density propagation. If the manifold has varying curvature, endpoints may act as sinks where the mass accumulates, acting as attractor points during reverse sampling (Shao et al. 2017). However, kernel density estimation may exaggerate these effects through smoothing and orientation-dependent artefacts. Scatter plots of raw samples

suggest that while the samples tend to the focus at the extremities, the KDE exaggerates the apparent tailing.

5.2 Image Generation in Latent Space

In this section, the CFFM model is applied to sample images from the *Smithsonian Butterflies Subset* dataset, rescaled to 128x128 pixels. Due to the curse of dimensionality, the application of multivariate KDEs in high-dimensions inhibits the effectiveness of the density approximation. To reduce these effects, dimensionality reduction via principle component analysis (PCA) is applied to the image datasets, allowing the model to be deployed in latent space.



(a) *Real Inversed PCA*

(b) *Generated Inversed PCA, KID = 0.148 ± 0.012*

(c) *Reduced PCA Inversed, KID = 0.1728 ± 0.0138*

Figure 5.5: Butterfly Generation (Hugging Face Hub n.d.). Comparison of butterfly image reconstruction. PCA decomposition was applied to all images, using 545 components. **(a)** Shows the real PCA reconstructed butterflies, images were reduced via the same PCA process and inversed, without application of the model. **(b)** Is the resultant images generated using the model, reconstructed included all original components. **(c)** To remove trailing PCA components (noise), 100 trailing components were removed to convey clearer image generation. Model application used $T = 200$, with annealed temperature and kernel bandwidth.

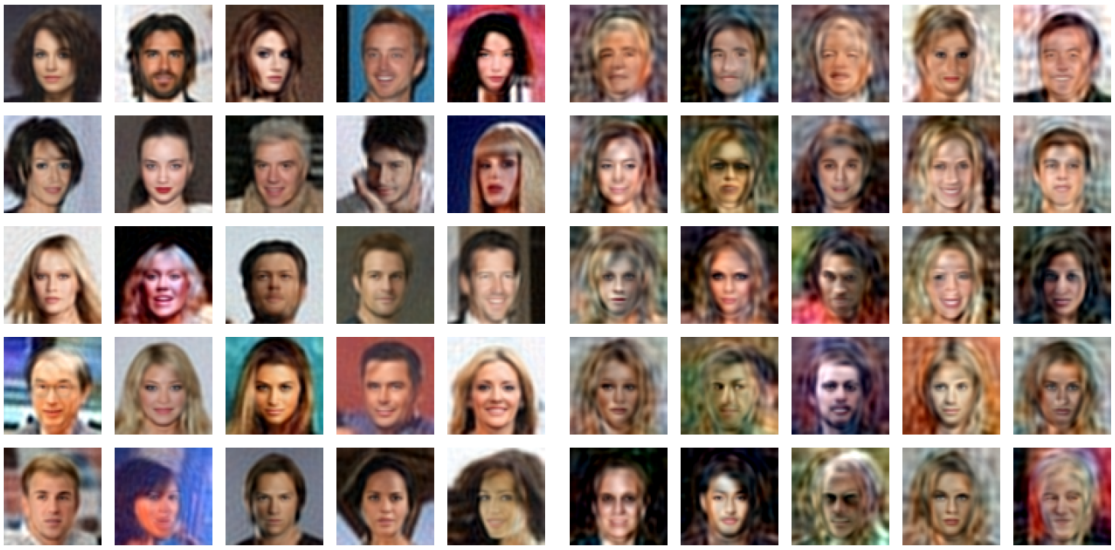
Figure 5.5, when applied to image data, the reconstructed butterflies in grid **(b)** exhibit well-defined structural features, accompanied by enhanced colour contrast. While the re-

constructions display evidence of a *global template* overlay characteristic from the dataset, distinct wing morphologies and fine-grained patterning remain discernible, such as colour differentiations around the wing tips, wing patterns, and venation. This indicates that meaningful underlying structure is preserved despite the global averaging effect. Grid **(c)** reduces residual artefacts by truncating trailing principal components, effectively suppressing high-frequency noise while retaining dominant structural information. The resulting images present smoother reconstructions with clearer wing outlines, though at the cost of reduced variability across samples. In contrast, the global template emerging from the KDE-based model amplifies dominant modes, observed in the oversaturated colours and darker samples, while suppressing finer morphological generation. The result produces homogenised outputs that differ qualitatively from the balanced reconstructions obtained via PCA alone.

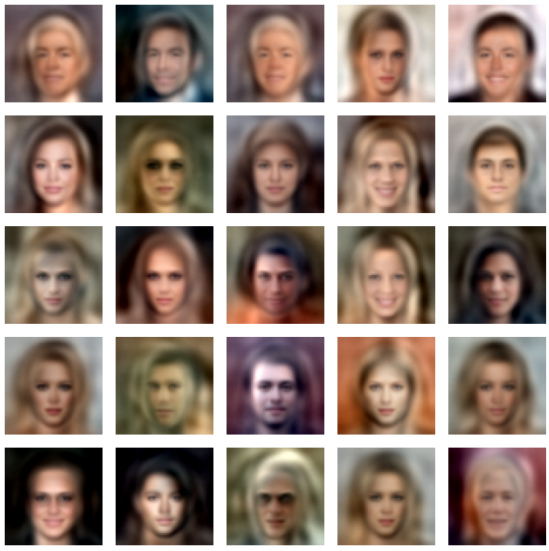
Application of the KDE-based sampling method to the image data demonstrates the implicit bias introduced by the kernel estimator. The global template can be interpreted as a high bias regime, where the kernel smoothing amplifies dominant modes of the distribution while suppressing rarer variations. This effect parallels the implicit bias observed in kernel based methods, with similar effects observed in Support Vector Machines, where choice of kernel bandwidth determines the decision boundaries smoothness, or sensitivity to fine-scale structure (Valentini & Dietterich 2004). On application of the PCA, the data distribution $p_{data}(x)$, compresses the butterflies into linear latent space, effectively linearising non-linear structure. Within this space, the KDEs from the terminal, source and potential functions define a scalar energy landscape. Langevin sampling then evolves trajectories along this surface, naturally favouring low-energy (high-density regions). The resultant density flow converges toward dominant modes, as observed in Figure 5.4, on application to the latent space, reconstructions that accentuate global temperature effects are produced, therefore diminishing rare variation. Hence, the model effectively recovers the manifold structure, attaining some of the intrinsic features of the butterfly, such as general colour, and faint patterns, but suppresses high frequency definition.

5.3 Facial Image Generation

To further test the models generation ability, we apply the model to latent space through PCA on the CelebA 64x64 dataset (Liu et al. 2015). While the Smithsonian Butterfly dataset provides clear colour and shape variations, the effect of the whitened background creates dominant variance due to a majority of the pixel space being occupied by white values, the latent directions associated with the background reduce semantic importance in colour and texture. This effect also results in the energy landscape flattening, in which the background acts as a flat basin where training points cluster. Application of the model to facial images, with non-uniform backgrounds, retains variance within the subject, most notably the hair, skin and lighting. Applying the model to this PCA space pushes the model toward semantic differences.



(a) *Real Faces (Reverted PCA)* **(b)** *Generated Face Images*



(c) *Reduced PCA*

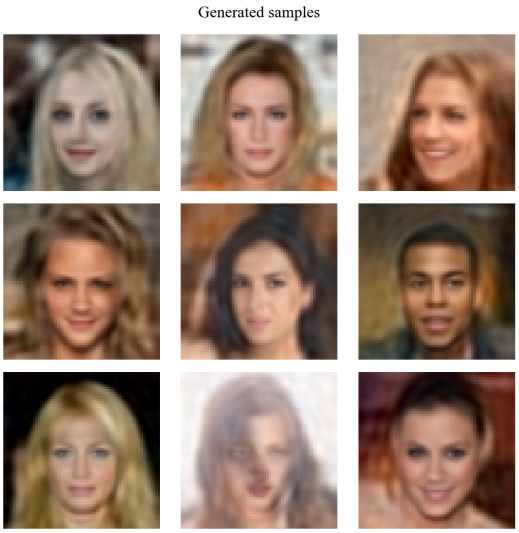
Figure 5.6: CelebA Generated Images. Comparison of real and generated (64x64) images under RBF definitions using a subset of the CelebA dataset (Liu et al. 2015). Grid **(a)** depicts the real faces, images were reduced using 545 PCA components and inversed to match conditions of generated images. Grid **(b)** shows the generated images, reduced to 545. Grid **(c)** displays the generated face images with trailing components removed.

The generated images in Figure 5.6 illustrate how variance concentrated around salient object features enables the model to capture finer structural detail. When trailing principal components are excluded, the resulting faces exhibit increased diversity. Despite a degree of blurring and feature superposition remaining, the reduction in background variance produces a more meaningful energy landscape, mitigating the presence of flattened basins, concentrat-

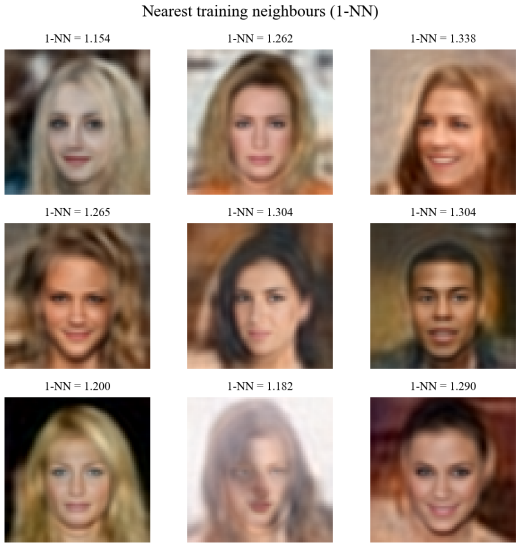
ing low energy regions on distinguishable features. The generated faces consistently preserve key attributes such as skin tone, mouth shape, eye structure, and minor lighting effects. However, the spatial positioning of the faces and the orientation of gaze remain largely central and outward-facing. This reflects a global templating effect, arising from the predominance of training data with aligned and front-facing configurations.

5.4 Ablation Studies

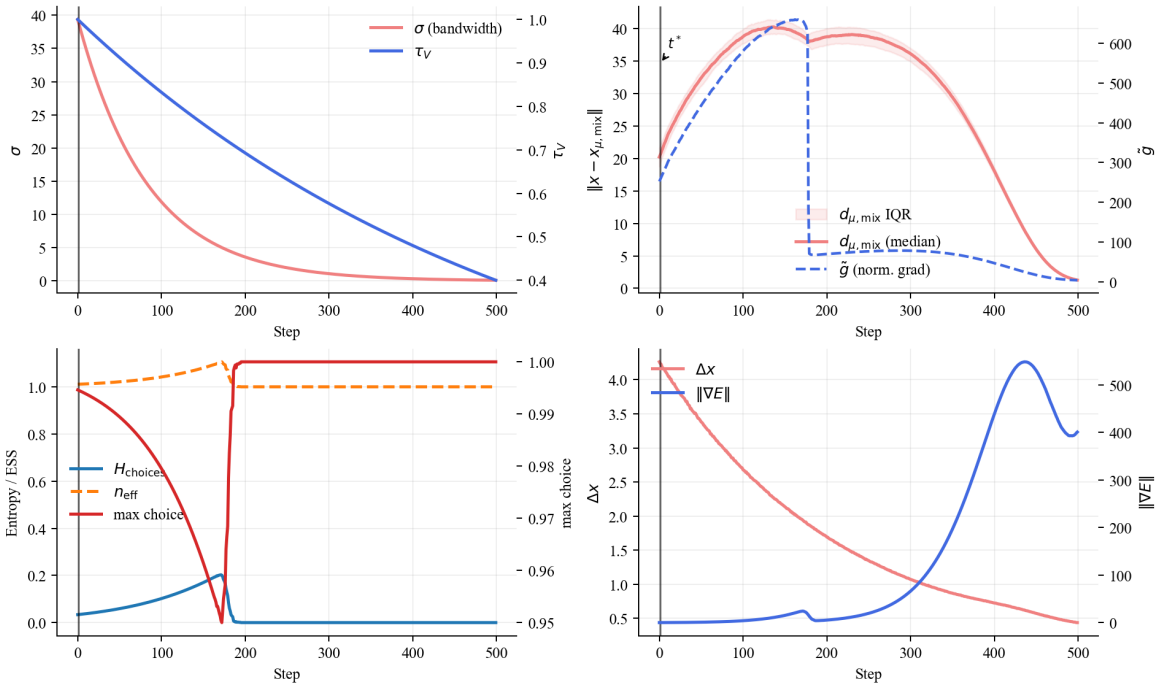
We study two operational modes of the kernel-driven reverse process. We compare the memorisation conditions, where low exploration, and a stronger terminal pull are applied. We also provide the conditions that increase generation, wherein higher exploration, slower temperature decay, with a weaker terminal pull are applied. The PCA space, paths L, and initial σ_0, and T sampling steps are held constant between simulations. The intention of this analysis is to distinguish between trajectories that collapse to training exemplars and those that support genuine generative behaviour.



(a) *Generated samples under memorisation settings.*



(b) *Nearest neighbours (1-NN) of the generated samples in the training set (PCA space).*



(c) *Diagnostics. Top-left: kernel bandwidth σ_t and potential temperature τ_V schedules. Top-right: median distance to mixture barycentre $d_{\mu,\text{mix}}$ (weighted average location of data under the kernel mixture), with interquartile range (pink) and schedule-normalised gradient $\tilde{g} = \sigma_t^2 \|\nabla E_t\|$ (dashed) (reflects the relative alignment of samples to the kernel landscape, not just the sharpening effect). Bottom-left: choice collapse statistics (entropy H_{choices}, effective set size n_{eff}, and maximum choice probability). Bottom-right: mean step displacement Δx_t and gradient norm $\|\nabla E_t\|$.*

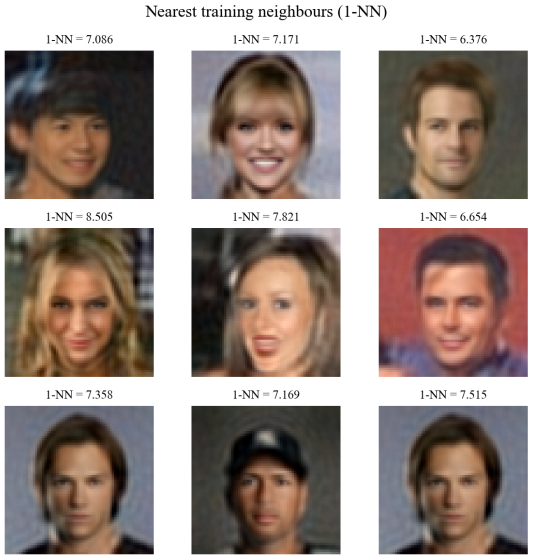
Figure 5.7: Memorisation regime. The model collapses toward training-set modes: generated samples (a) are visually close to their nearest neighbours (b), and diagnostics (c) show entropy collapse ($H_{\text{choices}} \to 0$), dominance of a single choice (max $\to 1$), shrinking $d_{\mu,\text{mix}}$, and steadily increasing Δx_t.

Figure 5.7 summarises the models dynamics under memorisation conditions. The top-left plot in **(c)** shows the annealing schedules for the kernel bandwidth σ_t and potential temperature τ_V, both of which decay monotonically. This sharpening produces narrower kernels and a sharper energy landscape, creating harder tessellations between close samples, reducing mixing across modes and biasing trajectories toward a single attractor. In the top-right panel, the median distance between samples and their mixture barycentre $d_{\mu,\text{mix}}$, together with its interquartile range (IQR), is plotted alongside the normalised gradient magnitude $\tilde{g}$. From $t = 0$ to $t \approx 175$, both $d_{\mu,\text{mix}}$ and $\tilde{g}$ increase steadily, while the narrow IQR indicates little dispersion between trajectories. At $t \approx 175$, $\tilde{g}$ collapses, followed by a sharp reduction in $d_{\mu,\text{mix}}$, signalling that trajectories have converged onto the barycentre of a single dominant mode, after which exploration ceases. The bottom-left plot further confirms this collapse, the entropy of the kernel choice distribution H_{choice} remains near zero, the effective sample size n_{eff} briefly spikes as trajectories switch between kernels but quickly returns to $n_{\text{eff}} = 1$, and the maximum kernel weight saturates, indicating that all samples are governed by the same mode. Finally, the bottom-right panel shows that the mean update size Δx decreases steadily as trajectories settle, while $\|\nabla_x E\|$ rises, peaking around $t \approx 450$ before declining, reflecting gradient saturation at the bottom of the energy well. Collectively, these diagnostics demonstrate that memorisation arises from rapid kernel sharpening and temperature decay, leading to an early collapse onto a single mode, vanishing entropy, diminishing updates as the sampler becomes trapped in one energy well.

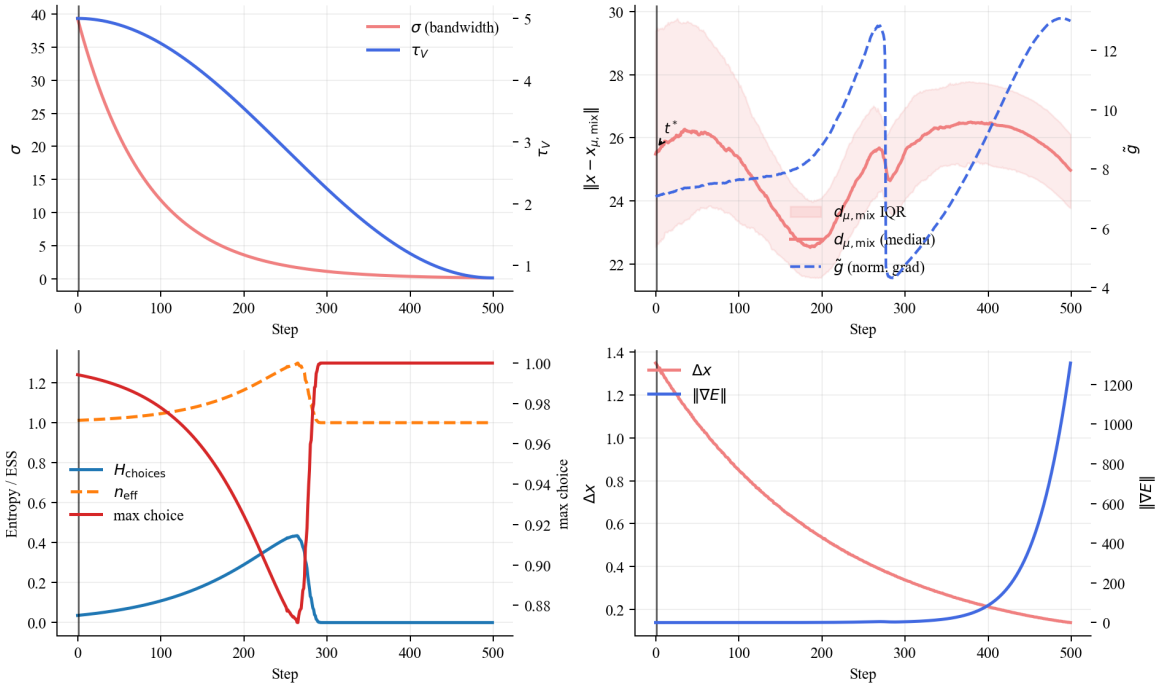
Generation Conditions

To induce generation conditions an alternative kernel scheme and temperature annealing is applied to encourage more global exploration. By increasing the bandwidth of the terminal condition, bias becomes more diffuse, covering a broader region, rather than collapsing to a single mode. The samples produced in 5.8 show visual distinction to their nearest neighbours. This detachment from the nearest samples is observed particularly through mouth and eye shaping. However, as observed in 5.6, higher levels of noise exist in the samples, demonstrating the generation-memorisation sharpness trade-off. Figure 5.8 shows this distinction, generated outputs align more broadly with regions of the dataset distribution, tending to, but not collapsing on specific exemplars.

(a) *Generated samples under high exploration*



(b) *Nearest neighbours (1-NN) of the generated samples in the training set (PCA space).*



(c) *Diagnostics. Top-left: kernel bandwidth σ_t and potential temperature τ_V schedules. Top-right: median distance to mixture barycentre $d_{\mu,\text{mix}}$ with IQR (pink) and schedule-normalised gradient $\tilde{g} = \sigma_t^2 \|\nabla E_t\|$ (dashed). Bottom-left: choice statistics showing higher entropy H_{choices}, larger effective set size n_{eff}, and lower maximum choice probability compared to memorisation. Bottom-right: step displacement Δx_t and gradient norm $\|\nabla E_t\|$, both remaining moderate and without collapse to a single mode.*

Figure 5.8: Generation regime. Samples (a) exhibit diversity and are not simple reconstructions of training examples, as confirmed by more distant 1-NN matches in (b). Diagnostic trends (c) indicate sustained exploration as indicated by the IQR width, and no sharp collapse into a singular mode is observed.

Figure 5.8 illustrates the sampler's behaviour under generation conditions. In this regime, the kernel bandwidth σ_t follows the same annealing schedule as in the memorisation case, but the potential temperature τ_V decays more slowly, maintaining smoother kernels for longer and thereby preserving stochasticity and exploration. The top right plot shows the barycentre mixture distance $d_{\mu,\text{mix}}$, which oscillates around a higher baseline with a wider IQR, indicating sustained trajectory dispersion, rather than collapse around modes. The normalised gradient $\tilde{g}$ remains small in magnitude compared to the memorisation regime, reflecting a smoother energy landscape. Around $t \approx 275$, $\tilde{g}$ undergoes a local collapse followed by a rapid monotonic increase, at a reduced scale in comparison to the memorisation scheme. This sequence suggests that while samples temporarily fall into wells, the smoother landscape allows them to escape, producing repeated cycles of collapse and recovery rather than permanent entrapment. The bottom-left reinforces this, as the entropy H_{choice} grows to higher levels and persists for longer, showing extended kernel mixing; n_{eff} exhibits larger and later spikes, indicating more effective neighbours contribute to the pull; and the collapse in maximum kernel choice is wider and shallower, spiking above 1.2 compared to $n_{eff} \approx 1.0$ in the memorisation regime. These features together imply a softer tessellation of the energy landscape, where no single kernel dominates immediately. Finally, in the bottom-right figure, the mean step size Δx decreases more slowly than under memorisation, while the raw gradient norm $\|\nabla_x E\|$ rises much more steeply, reaching significantly larger values by the end of sampling. This combination reflects trajectories continuing to move within broader basins rather than converging tightly to a single barycentre. Overall, generation conditions produce a smoother, less biased energy landscape, where trajectories predominantly favour one mode but remain spread across neighbouring kernels, leading to softer partitions and more diverse samples. This sustained exploration explains both the reduced collapse signals and the emergence of artefacts in generated outputs, as samples wander between overlapping regions instead of settling cleanly into single modes.

5.4.1 Computational Comparisons

This implementation of the method uses no neural parametrisations or back-propagation, which typically make neural networks computationally expensive. The main computational considerations are in the design of the gradient approximation. The method can be parallelised over multiple cores, however, in our implementation we focused on applying the model to a low resource setting. We deployed the model using a singular M path, ran on an Apple METAL M1 GPU. For query points, $X_t \in \mathbb{R}^{B\times L\times D}$, evaluated against a dataset, $X_{data} \in \mathbb{R}^{N\times D}$, the operations per gradient scales as,

$$O = N_{MC} \times B \times M \times N \times D$$

where N_{MC} denotes the number of Monte Carlo estimations used in the gradient approximation and B the batch size. Each kernel derivative effectively requires two passes over $O(BMND)$, due to the pair-wise distance calculation between all query points and the weighted barycentric sum via matrix multiplication. Since these operations are performed for the potential, source, and terminal kernels, the naïve implementation of the method requires an approximated $O(6BMND)$ FLOPs per Monte Carlo approximation. To compare

to other models in terms of computational requirements for gradient approximations, typical implementations are ordered as follows,

Model	Per-sample gradient evaluation complexity
CFFM (ours), $E(x,t)$	$O(N_{MC} \times B \times M \times N \times D)$
EBMs, $E_\theta(x)$	$O(P \times B)$
SGMs, $s_\theta(x,t)$	$O(P \times B)$
Flows, $v_\theta(x)$	$O(L \times B \times D \times h^2)$

Table 5.1: Comparison of gradient approximation complexity across models. Variables h and L represent the hidden layer size and layers respectively. The number of parameters is represented by P

Despite when initially observed, the complexity per gradient evaluation for EBMs and SGMs appears lesser, the parameter map P, even under small mappings is large. Under application to small toy data, P can be of magnitude $P \sim 10^5 - 10^6$, for larger models this easily reaches $10^6 - 10^9$. To demonstrate this, consider the models deployed in Figure 1.1, that underfit the data, in comparison to the CFFM model,

Model	Parameters P	Memory (FP32)
CFFM (ours)	0	< 8 KB (Store $X_{data} \in \mathbb{R}^{N \times D}$)
EBM	4×10^5	1.60 MB
SGM	4.5×10^5	1.82 MB
Flow	1.6×10^5	0.63 MB

Table 5.2: Parameter counts and storage requirements for generative models applied to the toy data. Model parameters and storage were derived from the models demonstrated in Figure 1.1 trained on toy data $X_{data} \in \mathbb{R}^{1000 \times 2}$

Unlike neural generative models, which require storage of the parameters themselves, our CFFM model requires no parameter storage. Instead, the data itself X_{data} acts as the memory, alongside kernel hyperparameters and schedules. This memory scales linearly with the size of the dataset, but remains interpretable as data storage, rather than model weight storage. In the naïve implementation, the CFFM model requires less storage, however, on application to large datasets, the model requires the memory of X_{data} for every evaluation, whereas neural networks are able to compress their representation of the data. Kernel approximation strategies (Nyström, random features, nearest-neighbour search) could reduce or remove this dependence, replacing N with a smaller effective basis size k where $k \ll N$. Thus, while the current implementation shows $O(6BMND)$, this is not a fundamental limitation of the method, but of the reference implementation.

Chapter 6

Discussion

The findings highlight how the model exhibits distinct behaviours on low-dimensional toy datasets, which serve as interpretable proxies for performance in higher-dimensional settings. In the Many Moons case (Figure 5.1), the impact of temperature on kernel weightings is particularly clear. At lower temperatures, sharper kernels produce a strong pull toward the manifold. Samples remain on the manifold but collapse toward the global mean, reflecting both the mean-reverting nature of the Ornstein–Uhlenbeck process (Popov et al. 2021) and the dominance of nearby kernels. At higher temperatures, smoother and more uniform weightings allow trajectories to spread across a larger portion of the manifold, promoting broader coverage and sustained exploration. However, this flexibility introduces artefacts as trajectories drift between overlapping regions.

By contrast, the chequerboard dataset (Figure 5.2) emphasises the model's ability to capture periodic, piecewise-linear structure. At low temperatures, sharp kernels strongly align particles to the manifold, yielding deeper trajectory penetration and more balanced mode coverage. At higher temperatures, the reduced discriminative power of smoother kernels causes probability mass to cluster near nodes closest to the initial distribution, while trajectories penetrate less deeply and exploration becomes limited. Together, these comparisons show that while lower temperatures enhance fidelity to local structure, they risk collapse through mean reversion, whereas higher temperatures support exploration but at the cost of artefacts or underfitting.

These observations motivated the temperature annealing strategy applied in Figure 5.3. By dynamically adjusting temperature across the sampling process, the model initially promotes global exploration, then progressively sharpens the kernels to emphasise lower-energy regions. This improves manifold coverage and mitigates premature collapse, producing more plausible samples.

From these findings, an informed design of kernel temperature schedules emerges, offering explicit control over whether the model operates in a generative or memorisation regime. In the memorisation setting, trajectories collapse deterministically onto single exemplars, yielding sharp but less diverse samples. In the generative setting, trajectories traverse multiple modes before stabilising, supporting diversity and interpolation at the cost of artefacts. The

temperature therefore provides an interpretable control knob mediating the trade-off between fidelity and diversity.

More broadly, these results reinforce the framing of the model as a closed-form flow model (CFFM), where the vector field is explicitly defined as the gradient of a scalar energy potential, $v(x,t) = -\nabla_x E(x,t)$. Unlike neural generative models that require learned parametrisations of this vector field, here the dynamics are governed directly by analytically specified kernels. This makes the impact of temperature and kernel weighting directly observable through barycentre distances, entropy, and gradient norms. The oscillations in $d_{\mu,\text{mix}}$ and spike of H_{choices} during generation can be interpreted as collapse-and-escape cycles, where trajectories orbit exemplar wells, showing momentary collapse into them before escaping through overlapping kernel regions when the landscape is sufficiently smooth. By contrast, in memorisation the sharper energy landscape prevents such escape, with gradients saturating as trajectories deterministically collapse. These dynamics illustrate how generation occurs not at the exemplars themselves, but in intermediate barycentric regions created by kernel overlaps, enabling interpolation to support novelty, but also introducing artefacts.

To extend these insights to higher-dimensional image data, dimensionality reduction was first applied using PCA to mitigate the curse of dimensionality in kernel evaluations. Based on the toy data results, stricter temperature scaling in the terminal kernel ($\tau_f = 0.1\kappa$) was used to induce memorisation, while more gradual schedules encouraged generative exploration. Through annealing, early iterations support broad exploration, while later iterations sharpen the kernels to guide trajectories toward low-energy regions. This regime improves manifold learning and yields more plausible image samples, demonstrating the application of closed-form flows beyond toy domains.

On consideration of the computational cost of implementing the model, the CFFM framework applied in this report achieves faster sampling times on small to medium datasets in comparison to SGMs, EBMs and Flow models. However, for larger datasets on limited hardware the memory requirement per gradient evaluation will be exceeded due to the linearity of model complexity, which will surpass that of the neural models.

Chapter 7

Conclusions

In this work we proposed a unified perspective of generative models as probability mass transport systems, in which Score-Based Diffusion Models, Energy-Based Models, and Flow Models can all be understood as specifying vector fields that govern generation. Within this framework we derived and implemented a non-parametric Feynman–Kac formulation with explicit bias and regularisation terms, in the form of the terminal and source functions, yielding a learning-free Closed-Form Flow Model. The design of these functions provides explicit control over the flows, with competitive, kernel-defined energy gradients modulating transport dynamics. Controlled smoothing and averaging were shown to yield either memorisation or novel generative behaviour, establishing both as outcomes of the same underlying transport mechanics.

While the CFFM demonstrates faster sampling on small to medium datasets as well as avoiding the heavy training costs of neural parametrisations, the implementation in this report scales linearly with dataset size. The requirements of the model in this report are $O(6MLND)$ FLOPs and full access to X_{data} at each gradient evaluation. This limits the scalability, however, the model provides clear advantages in low resource settings where interpretability outweighs raw fidelity. Such domain applications may include scientific simulation, where controllable bias and regularisation are valuable, as well as low resource or embedded settings where large neural networks are infeasible. Another application may be in benchmarking or teaching scenarios, where an analytically specified baseline is useful for studying the mechanics of generative modelling.

This report makes a novel contribution by introducing a unified framework for generative modelling, from which the Closed-Form Flow Model (CFFM) was derived. The resulting model is non-parametric and learning-free, enabling explicit investigation of the fidelity–diversity trade-off and providing a principled foundation for studying the structural dynamics of generative processes.

7.1 Limitations

While the computational limitations and accessibility of the model have been laid out in the previous chapters, this section focuses on theoretical limits of the current implementation.

Curse of Dimensionality

The proposed methodology provides a learning-free and interpretable generative framework suitable for low-dimensional settings, its extension to high-dimensional domains presents significant representational challenges. In such regimes, performance degradation manifests as either excessive smoothing or overfitting to training exemplars, observable through superposition artefacts and memorisation effects, respectively. These issues arise primarily from the reliance on Gaussian kernel density estimations that define the terminal, source, and potential functions. For application to higher dimensional domains, alternative kernels or descriptive functions should be implemented to overcome this.

Path Explorations

For efficiency, the reverse process was implemented with a single Feynman-Kac path ($L = 1$), exploiting the linear complexity of the formulation. This restriction ensured tractability on limited hardware while still producing viable generative outputs. However, the use of a single trajectory per particle limits the extent of exploration across the energy landscape, as no ensemble averaging is performed. Increasing the number of paths would permit each particle to traverse multiple stochastic trajectories, thereby reducing estimator variance and affording greater control over the balance between memorisation and generation. While this work prioritised a learning-free framework in low resource settings, future scaling of L offers a natural route to enhance generative diversity, with the caveat of increased computational complexity, that linearly scales with L.

Marginal Distribution Loss

Although not strictly required for generation, the current formulation does not preserve the forward marginals under Langevin dynamics. This breaks equivalence with SGMs, where marginal preservation is essential to guarantee that the reverse process produces samples consistent with the target data distribution. In SGMs, this is achieved by incorporating the gradients of the log-marginal densities (scores) of the forward process into the reverse dynamics, enabling noise to be progressively reshaped into realistic data. By contrast, if the process is defined as a deterministic reverse ODE, the marginals are preserved, thereby maintaining SGM equivalence. In the present formulation the framework can instead be interpreted in terms of flow-based models, since flows do not require intermediate marginals to be preserved The lack of this property does not fundamentally undermine generative performance, but rather shifts the interpretation away from the score-based paradigm towards the vector-field perspective.

Furthermore, the use of Langevin dynamics precludes direct computation of exact log-likelihoods under transformation. In general, when a random variable is transformed, the corresponding probability density (or mass) function must be derived via the change-of-variables theorem,

which introduces a Jacobian determinant for continuous variables, or probability ratio for discrete variables. This ensures that the transformed distribution is correctly normalised. The log-likelihood can then be obtained by evaluating the transformed variable under this new density and taking the logarithm. In the Langevin setting, however, the stochasticity of the updates prevents the straightforward application of this theorem, making exact likelihood evaluation intractable.

Solenoidal Flow Freedom

A further limitation arises from the restriction of the flow field to a purely conservative form, defined as $v_t = -\nabla_x E$. By construction, such flows are curl-free, and thus lack any solenoidal (divergence-free) component. This restriction removes potentially useful degrees of freedom that could, in principle, improve trajectory-level efficiency by allowing more flexible mass transport. However, the conservative formulation offers distinct advantages, it ensures global integrability, provides a clear energy-based interpretationn and guarantees that the induced score field is globally consistent. In this sense, the trade-off reflects a balance between expressive flexibility and interpretability.

Chapter 8

Future Work

8.0.1 Parametric Integration

A central design choice in this study was to retain a fully non-parametric formulation, prioritising interpretability and explicit control over the transport dynamics. While this approach yielded clear insights into the mechanisms underlying memorisation and generation, it inevitably limits scalability. KDEs suffer from the curse of dimensionality, which restricts the applicability of the current framework in higher-dimensional spaces. An immediate avenue for future work lies in exploring strategies that address this trade-off between interpretability and scalability.

One possibility is to remain strictly non-parametric but investigate methods that improve computational tractability. Examples include more efficient kernel approximations, low-rank embeddings, or adaptive bandwidth selection schemes that scale better in higher dimensions. Such extensions would preserve the fully transparent structure of the current framework while mitigating the dimensionality bottleneck.

A second direction involves partial parametrisation of the framework. For instance, Scarvelis et al. (2025) embed the data space into a lower-dimensional latent representation VAE, with the non-parametric reverse process then applied in this compressed space. This would allow the non-parametric transport mechanics to act on a more manageable latent manifold, while the decoder reintroduces data fidelity. Such a hybrid approach would allow the exploration of how the memorisation–generation dynamics, observed in this study, to carry over when data is represented in parametric latent spaces.

Finally, a more ambitious route would be to fully parametrise the components of the model itself. For example, the kernel functions could be replaced with neural approximators, while preserving their structural roles as source, terminal, and potential terms. This would enable scaling to high-dimensional datasets while retaining the theoretical grounding in energy transport. Crucially, the diagnostic quantities derived in this work (such as barycentre distances, choice entropy, effective set size, schedule-normalised gradients) could be repurposed as regularisers or interpretability metrics for training parametric generative models such as Diffusion Models or EBMs. This would allow a direct test of whether the mechanisms highlighted in this work, namely the role of bandwidth sharpening, temperature schedules,

and kernel overlaps, mediate the memorisation–generation trade-off, which is also present in large-scale neural generative systems.

In this way, the present study can be viewed as a prototype for a wider research programme. Starting from a fully interpretable, non-parametric baseline, future work could progressively introduce parametrisation to balance interpretability, scalability, and generative capability.

The ultimate objective would be to examine whether insights from kernel-driven transport dynamics can improve both the performance and the interpretability of contemporary parametric generative models.

8.0.2 Jump Diffusions

A promising direction for future research involves extending the current framework to incorporate jump diffusion processes through the Feynman-Kac formula. Recent evidence suggests that state-of-the-art generative models exhibit improved generalisation when subjected to heavy-tailed noise injection (Baule 2025, Pandey et al. 2024), motivating the exploration of non-Gaussian dynamics in generative modelling.

Jump diffusion processes, characterised by continuous stochastic evolution punctuated by discrete random events (analogous to neuronal firing patterns), offer several theoretical advantages over standard Brownian motion. Specifically, jump processes enable more efficient exploration of multimodal distributions through their inherent ability to traverse energy barriers via large, instantaneous transitions. While conventional diffusion processes generate Gaussian-tailed distributions, jump diffusions naturally produce mixture distributions with heavy tails due to rare but significant deviations introduced by the jump component (Shariatian et al. 2025).

The integration of jump dynamics into the current Feynman-Kac framework could be formulated through the following stochastic differential equation,

$$dX_t = f(X_t, t)dt + g(X_t, t)dW_t + J_t\, dN_t \tag{8.1}$$

Where N_t is a Poisson counting process with rate λ, and J_t is the associated jump size drawn from some distribution (Øksendal & Sulem 2005). Where previously under the regular Brownian SDE seen in Equation 2.10 trajectories of particles were continuous, under jump diffusions the system evolves continuously most of the time, but at random jump times (with expected frequency λ), the state is displaced by J_t. These discontinuous updates allow trajectories to bypass local energy barriers, making exploration of multimodal landscapes more efficient than with Gaussian noise alone.

A particularly intriguing extension involves incorporating the Feynman-Kac solution, $u(x, t)$, directly into the jump mechanism. By making the jump intensity λ, adaptive to the local potential landscape, specifically,

$$\lambda(x, t) = h(u(X_t, t)) = h(E(X_t, t))$$

in which h is some arbitrary function. The law of the process is dependent on the expected energy functional along the trajectories. This results in the system becoming a McKean-Vlasov jump diffusion,

$$dX_t = f(X_t, t)dt + g(X_t, t)dW_t + J(E(X_t, t))\, dN_t \tag{8.2}$$

with $\mu_t = \text{Law}(X_t)$ entering implicitly through $E(x, t)$. This makes jumps adaptable to where the probability mass flows, making the system non-linear in distribution (Rehmeier & Röckner 2024). The importance of this extension lies in scalability, such processes admit mean-field limits under propagation of chaos, effectively approximating infinite-dimensional particle systems with tractable stochastic dynamics (Agram & Rems 2025). This provides a principled route to model long-term, collective behaviour in generative systems, overcoming the curse of dimensionality that constrains kernel-based methods. Recent neural approaches have begun to approximate these processes in parametric form. However, incorporating them analytically through the CFFM framework would allow direct interpretability while retaining scalability. Therefore, this would enable more efficient high-dimensional generative modelling, richer time-series dynamics, and new ways of inferring system parameters directly from data (Yang, Hasan, Ng & Tarokh 2024).

Bibliography

Agram, N. & Rems, J. (2025), 'Deep learning for conditional mckean–vlasov jump diffusions', *Systems Control Letters* **201**, 106100.
URL: *https://www.sciencedirect.com/science/article/pii/S0167691125000829*

Aithal, S. K., Maini, P., Lipton, Z. C. & Kolter, J. Z. (2024), 'Understanding hallucinations in diffusion models through mode interpolation'.
URL: *https://arxiv.org/abs/2406.09358*

Anderson, B. D. (1982), 'Reverse-time diffusion equation models', *Stochastic Processes and their Applications* **12**(3), 313–326.
URL: *https://www.sciencedirect.com/science/article/pii/0304414982900515*

Balcerak, M., Amiranashvili, T., Terpin, A., Shit, S., Bogensperger, L., Kaltenbach, S., Koumoutsakos, P. & Menze, B. (2025), 'Energy matching: Unifying flow matching and energy-based models for generative modeling'.
URL: *https://arxiv.org/abs/2504.10612*

Baule, A. (2025), 'Generative modelling with jump-diffusions'.
URL: *https://arxiv.org/abs/2503.06558*

Bettinger, J. S. & Friston, K. J. (2023), 'Conceptual foundations of physiological regulation incorporating the free energy principle and self-organized criticality', *Neuroscience Biobehavioral Reviews* **155**, 105459.
URL: *https://www.sciencedirect.com/science/article/pii/S0149763423004281*

Bugmann, G. (1998), 'Normalized gaussian radial basis function networks', *Neurocomputing* **20**(1), 97–110.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231298000277*

Bösch, C., Roeder, G., Serra-Garcia, M. & Adams, R. P. (2025), 'Local learning rules for out-of-equilibrium physical generative models'.
URL: *https://arxiv.org/abs/2506.19136*

Chang, S., Cohen, T. & Ostdiek, B. (2018), 'What is the machine learning?', *Phys. Rev. D* **97**, 056009.
URL: *https://link.aps.org/doi/10.1103/PhysRevD.97.056009*

Chen, M., Huang, K., Zhao, T. & Wang, M. (2023), 'Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data'.
URL: *https://arxiv.org/abs/2302.07194*

Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. (2019), 'Neural ordinary differential equations'.
URL: *https://arxiv.org/abs/1806.07366*

Chen, S., Chewi, S., Lee, H., Li, Y., Lu, J. & Salim, A. (2023), 'The probability flow ode is provably fast'.
URL: *https://arxiv.org/abs/2305.11798*

Daras, G., Chung, H., Lai, C.-H., Mitsufuji, Y., Ye, J. C., Milanfar, P., Dimakis, A. G. & Delbracio, M. (2024), 'A survey on diffusion models for inverse problems'.
URL: *https://arxiv.org/abs/2410.00083*

Del Moral, P. (2004), *Feynman-Kac formulae : genealogical and interacting particle systems with applications*, Probability and its applications, Springer, New York ;.

Deveney, T., Stanczuk, J., Kreusser, L. M., Budd, C. & Schönlieb, C.-B. (2023), 'Closing the ode-sde gap in score-based diffusion models through the fokker-planck equation'.
URL: *https://arxiv.org/abs/2311.15996*

Esser, P., Chiu, J., Atighehchian, P., Granskog, J. & Germanidis, A. (2023), Structure and content-guided video synthesis with diffusion models, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 7346–7356.

Fang, Z., Díaz, M., Buchanan, S. & Sulam, J. (2025), 'Beyond scores: Proximal diffusion models'.
URL: *https://arxiv.org/abs/2507.08956*

Friston, K. (2010), 'The free-energy principle: a unified brain theory?', *Nature Reviews Neuroscience* **11**(2), 127–138.

Friston, K., Da Costa, L., Sajid, N., Heins, C., Ueltzhöffer, K., Pavliotis, G. A. & Parr, T. (2023), 'The free energy principle made simpler but not too simple', *Physics Reports* **1024**, 1–29. The free energy principle made simpler but not too simple.
URL: *https://www.sciencedirect.com/science/article/pii/S037015732300203X*

Gao, Y., Huang, J., Jiao, Y. & Zheng, S. (2024), 'Convergence of continuous normalizing flows for learning probability distributions'.
URL: *https://arxiv.org/abs/2404.00551*

Geng, C., Han, T., Jiang, P.-T., Zhang, H., Chen, J., Hauberg, S. & Li, B. (2024), 'Improving adversarial energy-based model via diffusion process'.
URL: *https://arxiv.org/abs/2403.01666*

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), 'Generative adversarial networks'.
URL: *https://arxiv.org/abs/1406.2661*

Google DeepMind (2025), 'Veo 3 model card', https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf. Published May 23, 2025.

Gutmann, M. & Hyvärinen, A. (2010), Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, *in* Y. W. Teh & M. Titterington, eds, 'Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics', Vol. 9 of *Proceedings of Machine Learning Research*, PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 297–304.
URL: *https://proceedings.mlr.press/v9/gutmann10a.html*

Hinton, G. (2017), Boltzmann machines, *in* 'Encyclopedia of machine learning and data mining', Springer, pp. 164–168.

Hinton, G. E. (2002), 'Training products of experts by minimizing contrastive divergence', *Neural Computation* **14**(8), 1771–1800.

Ho, J., Jain, A. & Abbeel, P. (2020), 'Denoising diffusion probabilistic models'.
URL: *https://arxiv.org/abs/2006.11239*

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M. & Fleet, D. J. (2022), 'Video diffusion models'.
URL: *https://arxiv.org/abs/2204.03458*

Huembeli, P., Arrazola, J. M., Killoran, N., Mohseni, M. & Wittek, P. (2022), 'The physics of energy-based models', *Quantum Machine Intelligence* **4**(1), 1.
URL: *https://doi.org/10.1007/s42484-021-00057-7*

Hugging Face Hub (n.d.), 'Smithsonian butterflies subset', https://huggingface.co/datasets/huggan/smithsonian$_butterflies_subset$. *Accessed* : $2025-08-26$.

Hyvärinen, A. (2005), 'Estimation of non-normalized statistical models by score matching', *Journal of Machine Learning Research* **6**(24), 695–709.
URL: *http://jmlr.org/papers/v6/hyvarinen05a.html*

Jaynes, E. T. (1957), 'Information theory and statistical mechanics', *Phys. Rev.* **106**, 620–630.
URL: *https://link.aps.org/doi/10.1103/PhysRev.106.620*

Jordan, R., Kinderlehrer, D. & Otto, F. (1997), 'Free energy and the fokker-planck equation', *Physica D: Nonlinear Phenomena* **107**(2), 265–271. 16th Annual International Conference of the Center for Nonlinear Studies.
URL: *https://www.sciencedirect.com/science/article/pii/S0167278997000936*

Ju, Z., Wang, Y., Shen, K., Tan, X., Xin, D., Yang, D., Liu, Y., Leng, Y., Song, K., Tang, S., Wu, Z., Qin, T., Li, X.-Y., Ye, W., Zhang, S., Bian, J., He, L., Li, J. & Zhao, S. (2024), 'Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models'.
URL: *https://arxiv.org/abs/2403.03100*

Jumper, J., Evans, R., Pritzel, A. et al. (2021), 'Highly accurate protein structure prediction with alphafold', *Nature* **596**(7873), 583–589.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y. & Ermon, S. (2023), 'Consistency trajectory models: Learning probability flow ode trajectory of diffusion', *arXiv preprint arXiv:2310.02279* .

Kim, D., Na, B., Kwon, S. J., Lee, D., Kang, W. & Moon, I.-C. (2022), 'Maximum likelihood training of implicit nonlinear diffusion models'.
URL: *https://arxiv.org/abs/2205.13699*

Kingma, D. P. & LeCun, Y. (2010), Regularized estimation of image statistics by score matching, *in* 'Proceedings of the 24th International Conference on Neural Information Processing Systems - Volume 1', NIPS'10, Curran Associates Inc., Red Hook, NY, USA, p. 1126–1134.

Kingma, D. P. & Welling, M. (2022), 'Auto-encoding variational bayes'.
URL: *https://arxiv.org/abs/1312.6114*

Kobyzev, I., Prince, S. J. & Brubaker, M. A. (2021), 'Normalizing flows: An introduction and review of current methods', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(11), 3964–3979.
URL: *http://dx.doi.org/10.1109/TPAMI.2020.2992934*

Koller, D. & Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, MA.
URL: *https://books.google.co.uk/books?id=7dzpHCHzNQ4C*

Labs, I., Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., Ermon, S., Grover, A. & Kuleshov, V. (2025), 'Mercury: Ultra-fast language models based on diffusion'.
URL: *https://arxiv.org/abs/2506.17298*

Lai, C.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y. & Ermon, S. (2023), 'Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation'.
URL: *https://arxiv.org/abs/2210.04296*

Lecun, Y., Chopra, S. & Hadsell, R. (2006), 'A tutorial on energy-based learning'.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F. et al. (2006), 'A tutorial on energy-based learning', *Predicting structured data* **1**(0).

Li, X., Dai, Y. & Qu, Q. (2024), 'Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure'.
URL: *https://arxiv.org/abs/2410.24060*

Lin, J. (2012), 'Backward stochastic differential equations and feynman-kac formula for multidimensional lévy processes, with applications in finance'.
URL: *https://arxiv.org/abs/1201.6614*

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M. & Le, M. (2023), 'Flow matching for generative modeling'.
URL: *https://arxiv.org/abs/2210.02747*

Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H. & Gat, I. (2024), 'Flow matching guide and code', *arXiv preprint arXiv:2412.06264* .

Liu, M., Shi, J., Cao, K., Zhu, J. & Liu, S. (2017), 'Analyzing the training processes of deep generative models', *IEEE transactions on visualization and computer graphics* **24**(1), 77–87.

Liu, Z., Luo, P., Wang, X. & Tang, X. (2015), Deep learning face attributes in the wild, *in* 'Proceedings of International Conference on Computer Vision (ICCV)'.

Lyu, S. (2012), 'Interpretation and generalization of score matching'.
URL: *https://arxiv.org/abs/1205.2629*

Ma, Y. & Fu, Y., eds (2011), *Manifold Learning Theory and Applications*, 1st edn, CRC Press.

Ma, Z., Zhang, Y., Jia, G., Zhao, L., Ma, Y., Ma, M., Liu, G., Zhang, K., Li, J. & Zhou, B. (2024), 'Efficient diffusion models: A comprehensive survey from principles to practices'.
URL: *https://arxiv.org/abs/2410.11795*

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R. & Li, C. (2025), 'Large language diffusion models'.
URL: *https://arxiv.org/abs/2502.09992*

Niedoba, M., Zwartsenberg, B., Murphy, K. & Wood, F. (2025), 'Towards a mechanistic explanation of diffusion model generalization'.
URL: *https://arxiv.org/abs/2411.19339*

Øksendal, B. (2003), Stochastic differential equations, *in* 'Stochastic differential equations: an introduction with applications', Springer, pp. 38–50.

Orr, M. J. et al. (1996), 'Introduction to radial basis function networks'.

Ou, Z. (2024), 'Energy-based models with applications to speech and language processing', *Foundations and Trends® in Signal Processing* **18**(1–2), 1–199.
URL: *http://dx.doi.org/10.1561/2000000117*

Pandey, K., Pathak, J., Xu, Y., Mandt, S., Pritchard, M., Vahdat, A. & Mardani, M. (2024), 'Heavy-tailed diffusion models', *arXiv preprint arXiv:2410.14171* .

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. & Lakshminarayanan, B. (2021), 'Normalizing flows for probabilistic modeling and inference'.
URL: *https://arxiv.org/abs/1912.02762*

Pardoux, E. & Peng, S. (1992), Backward stochastic differential equations and quasilinear parabolic partial differential equations, *in* B. L. Rozovskii & R. B. Sowers, eds, 'Stochastic Partial Differential Equations and Their Applications', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 200–217.

Parisi, G. (1981), 'Correlation functions and computer simulations', *Nuclear Physics B* **180**(3), 378–384.
URL: *https://www.sciencedirect.com/science/article/pii/0550321381900560*

Pham, H. (2014), 'Feynman-kac representation of fully nonlinear pdes and applications'.
URL: *https://arxiv.org/abs/1409.0625*

Pidstrigach, J. (2022), 'Score-based generative models detect manifolds'.
URL: *https://arxiv.org/abs/2206.01018*

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T. & Kudinov, M. (2021), 'Grad-tts: A diffusion probabilistic model for text-to-speech'.
URL: *https://arxiv.org/abs/2105.06337*

Rehmeier, M. & Röckner, M. (2024), 'On nonlinear markov processes in the sense of mckean'.
URL: *https://arxiv.org/abs/2212.12424*

Ross, B. L., Kamkari, H., Wu, T., Hosseinzadeh, R., Liu, Z., Stein, G., Cresswell, J. C. & Loaiza-Ganem, G. (2025), 'A geometric framework for understanding memorization in generative models'.
URL: *https://arxiv.org/abs/2411.00113*

Salimans, T. & Ho, J. (2022), 'Progressive distillation for fast sampling of diffusion models'.
URL: *https://arxiv.org/abs/2202.00512*

Scarvelis, C., de Ocáriz Borde, H. S. & Solomon, J. (2025), 'Closed-form diffusion models'.
URL: *https://arxiv.org/abs/2310.12395*

Shao, H., Kumar, A. & Fletcher, P. T. (2017), 'The riemannian geometry of deep generative models'.
URL: *https://arxiv.org/abs/1711.08014*

Shariatian, D., Simsekli, U. & Durmus, A. (2025), 'Heavy-tailed diffusion with denoising lévy probabilistic models'.
URL: *https://arxiv.org/abs/2407.18609*

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M. & Tang, J. (2020), 'Graphaf: a flow-based autoregressive model for molecular graph generation', *arXiv preprint arXiv:2001.09382* .

Shi, Z., Zhou, X., Qiu, X. & Zhu, X. (2020), 'Improving image captioning with better use of captions'.
URL: *https://arxiv.org/abs/2006.11807*

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N. & Ganguli, S. (2015), 'Deep unsupervised learning using nonequilibrium thermodynamics'.
URL: *https://arxiv.org/abs/1503.03585*

Song, Y. & Ermon, S. (2020), 'Generative modeling by estimating gradients of the data distribution'.
URL: *https://arxiv.org/abs/1907.05600*

Song, Y., Garg, S., Shi, J. & Ermon, S. (2020), Sliced score matching: A scalable approach to density and score estimation, *in* R. P. Adams & V. Gogate, eds, 'Proceedings of The 35th Uncertainty in Artificial Intelligence Conference', Vol. 115 of *Proceedings of Machine Learning Research*, PMLR, pp. 574–584.
URL: *https://proceedings.mlr.press/v115/song20a.html*

Song, Y. & Kingma, D. P. (2021), 'How to train your energy-based models'.
URL: *https://arxiv.org/abs/2101.03288*

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. & Poole, B. (2021), 'Score-based generative modeling through stochastic differential equations'.
URL: *https://arxiv.org/abs/2011.13456*

Tong, A., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G. & Bengio, Y. (2024), 'Simulation-free schrödinger bridges via score and flow matching'.
URL: *https://arxiv.org/abs/2307.03672*

Valentini, G. & Dietterich, T. (2004), 'Bias-variance analysis of support vector machines for the development of svm-based ensemble methods.'.

Wang, C. (2025), 'Wasserstein metric, gradient flow structure and well-posedness of fokker-planck equation on locally finite graphs'.
URL: *https://arxiv.org/abs/2503.03531*

Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W. & Zhou, M. (2023), 'Patch diffusion: Faster and more data-efficient training of diffusion models'.
URL: *https://arxiv.org/abs/2304.12526*

Xu, M., Geffner, T., Kreis, K., Nie, W., Xu, Y., Leskovec, J., Ermon, S. & Vahdat, A. (2025), 'Energy-based diffusion language models for text generation'.
URL: *https://arxiv.org/abs/2410.21357*

Yang, H., Hasan, A., Ng, Y. & Tarokh, V. (2024), 'Neural mckean-vlasov processes: Distributional dependence in diffusion processes'.
URL: *https://arxiv.org/abs/2404.09402*

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B. & Yang, M.-H. (2024), 'Diffusion models: A comprehensive survey of methods and applications'.
URL: *https://arxiv.org/abs/2209.00796*

Yi, M., Sun, J. & Li, Z. (2023), 'On the generalization of diffusion model'.
URL: *https://arxiv.org/abs/2305.14712*

Younes, L. (1999), 'On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates', *Stochastics and Stochastic Reports* **65**(3-4), 177–228.
URL: *https://doi.org/10.1080/17442509908834179*

Yu, L., Song, Y., Song, J. & Ermon, S. (2020), 'Training deep energy-based models with f-divergence minimization'.
URL: *https://arxiv.org/abs/2003.03463*

Zang, C. & Wang, F. (2020), Moflow: an invertible flow model for generating molecular graphs, *in* 'Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining', pp. 617–626.

Zhang, H., Zhou, J., Lu, Y., Guo, M., Wang, P., Shen, L. & Qu, Q. (2024), 'The emergence of reproducibility and generalizability in diffusion models'.
URL: *https://arxiv.org/abs/2310.05264*

Øksendal, B. & Sulem, A. (2005), *Applied Stochastic Control of Jump Diffusions*, Universitext, 2nd edn, Springer.

Appendices

Appendix A

Appendix

A.1 Derivations

A.1.1 NLL EBM Derivation

We begin with the definition of the NLL under the data distribution $p_{\text{data}}(\mathbf{x})$,

$$-\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log p_\theta(\mathbf{x})] = -\int p_{\text{data}}(\mathbf{x}) \log p_\theta(\mathbf{x})\, d\mathbf{x} \tag{A.1}$$

Now recall the definition of the Kullback–Leibler (KL) divergence between $p_{\text{data}}(\mathbf{x})$ and $p_\theta(\mathbf{x})$:

$$D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \,\|\, p_\theta(\mathbf{x})) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\, d\mathbf{x} \tag{A.2}$$

We can expand this expression using the logarithm rule $\log \frac{a}{b} = \log a - \log b$:

$$D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) = \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x})\, d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_\theta(\mathbf{x})\, d\mathbf{x} \tag{A.3}$$

Recognising the expectations, this becomes,

$$D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_{\text{data}}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] \tag{A.4}$$

Rearranging terms,

$$-\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] = D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_{\text{data}}(\mathbf{x})] \tag{A.5}$$

Therefore, the expected negative log-likelihood is equal to the KL divergence plus a constant (since the second term is independent of θ),

$$-\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] = D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta(\mathbf{x})) + \text{const} \tag{A.6}$$

A.1.2 Intractable Gradient of the Log Partition Function

First, begin by expanding the negative gradient of the log-likelihood for an energy-based model,

$$\begin{aligned} -\nabla_\theta \log p_\theta(\mathbf{x}) &= -\nabla_\theta \left(-E_\theta(\mathbf{x}) - \log Z_\theta\right) \\ &= \nabla_\theta E_\theta(\mathbf{x}) + \nabla_\theta \log Z_\theta \end{aligned} \tag{A.7}$$

To evaluate the intractable term $\nabla_\theta \log Z_\theta$, we apply the logarithmic identity,

$$\nabla_\theta \log Z_\theta = \frac{\nabla_\theta Z_\theta}{Z_\theta} \tag{A.8}$$

Now consider the definition of the partition function,

$$Z_\theta = \int e^{-E_\theta(\mathbf{x})}\, d\mathbf{x} \tag{A.9}$$

Assuming $E_\theta(\mathbf{x})$ is continuous in θ and that there exists an integrable function $g(\mathbf{x})$ such that

$$\left|\nabla_\theta e^{-E_\theta(\mathbf{x})}\right| \leq g(\mathbf{x}) \quad \text{for all } \theta,$$

we can apply Leibniz's Rule to interchange the gradient and the integral,

$$\nabla_\theta Z_\theta = \int \nabla_\theta e^{-E_\theta(\mathbf{x})}\, d\mathbf{x} \tag{A.10}$$

Using the chain rule,

$$\nabla_\theta e^{-E_\theta(\mathbf{x})} = -e^{-E_\theta(\mathbf{x})} \nabla_\theta E_\theta(\mathbf{x}) \tag{A.11}$$

Leading to the expression,

$$\nabla_\theta Z_\theta = -\int \nabla_\theta E_\theta(\mathbf{x})\, e^{-E_\theta(\mathbf{x})}\, d\mathbf{x} \tag{A.12}$$

Substitute this result into the earlier expression for $\nabla_\theta \log Z_\theta$,

$$\begin{aligned} \nabla_\theta \log Z_\theta &= \frac{1}{Z_\theta} \left(-\int \nabla_\theta E_\theta(\mathbf{x})\, e^{-E_\theta(\mathbf{x})}\, d\mathbf{x}\right) \\ &= -\int \nabla_\theta E_\theta(\mathbf{x})\, \frac{e^{-E_\theta(\mathbf{x})}}{Z_\theta}\, d\mathbf{x} \\ &= -\int \nabla_\theta E_\theta(\mathbf{x})\, p_\theta(\mathbf{x})\, d\mathbf{x} \end{aligned} \tag{A.13}$$

Which is equivalent to,

$$\nabla_\theta \log Z_\theta = -\mathbb{E}_{\mathbf{x}\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{x})] \tag{A.14}$$

Substituting this into the earlier expression for the negative log-likelihood gradient,

$$\begin{aligned} -\nabla_\theta \log p_\theta(\mathbf{x}) &= \nabla_\theta E_\theta(\mathbf{x}) + \nabla_\theta \log Z_\theta \\ &= \nabla_\theta E_\theta(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{x})] \end{aligned} \tag{A.15}$$

This final expression shows that the gradient of the negative log-likelihood decomposes into a positive energy gradient at the data point and a negative expected energy gradient under the model distribution.

A.1.3 Score Matching

Given general functions $f(x)$ and $g(x)$ are real and continuous differentiable functions with equal first derivatives everywhere, then it can be assumed $f(x) \equiv g(x)+C$ (Song & Kingma 2021). If the general functions $f(x)$ and $g(x)$ are expressed through PDFs,$p(x)$ and $q(x)$ respectively, we know that from the normalisation requirement in Equation (1.1) that,

$$\int e^{p(\mathbf{x})}\, d\mathbf{x} = \int e^{q(\mathbf{x})}\, d\mathbf{x} = 1 \tag{A.16}$$

Therefore it is derived that $f(x) \equiv g(x)$. This allows the approximation for learning the EBM through matching the first derivative of its log-PDF to the log-PDF if the data distribution. This first order gradient of the EBMs log-PDF is referred to as the *score* of the PDF, since it measures the difference between the two true and estimated distribution. Through this framing, by taking the first derivative of both PDFs w.r.t. the data, $\mathbf{x}$, the partition function is rendered as a constant, since it has no dependence on the data,

$$\Rightarrow -\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} E_\theta(\mathbf{x}) + \underbrace{\nabla_{\mathbf{x}} \log Z_\theta}_{=0} \tag{A.17}$$

Leading to the general expression,

$$\Rightarrow -\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} E_\theta(\mathbf{x}) \tag{A.18}$$

The Score Matching objective (Hyvärinen 2005) can now be deployed through minimising the dependency between the data distribution and parametrised distribution through the expression found in Equation (1.4). However, since the score function only relies upon the energy, rather than minimising the KL Divergence, we instead minimise the Fisher Divergence, which compares the log densities (scores),

$$D_{\mathrm{F}}(p_{\mathrm{data}}(\mathbf{x}) \,\|\, p_\theta(\mathbf{x})) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\frac{1}{2}||\nabla_{\mathbf{x}} \log p_{data}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})||^2\right] \tag{A.19}$$

However, this objective implies we must know the data distributions score function, $s_{data}(\mathbf{x})$, from the observed sample, introducing a non-parametric estimation problem. This can be avoided by applying integration by parts to the Fisher Divergence (**??**),

$$D_{\mathrm{F}}(p_{\mathrm{data}}(\mathbf{x}) \,\|\, p_\theta(\mathbf{x})) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\partial E_\theta(\mathbf{x})}{\partial \mathbf{x_i}}\right)^2 + \frac{\partial^2 E_\theta(\mathbf{x})}{\partial {\mathbf{x_i}}^2}\right] + \text{constant} \tag{A.20}$$

The unknown derivative, $\nabla_{\mathbf{x}} \log p_{data}(\mathbf{x})$, is now replaced, and d is the dimensionality of $\mathbf{x} \in \mathbb{R}^d$. The integration constant can be dropped in training since it has no dependence on the data, hence does not effect optimisation,

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\frac{1}{2}||\nabla_{\mathbf{x}} E_\theta(\mathbf{x})||^2 - \Delta_{\mathbf{x}} E_\theta(\mathbf{x})\right] \tag{A.21}$$

Where,

$$\Delta_{\mathbf{x}}\, E_\theta(\mathbf{x}) = \sum_i \frac{\partial^2 E_\theta(\mathbf{x})}{\partial {\mathbf{x_i}}^2} \tag{A.22}$$

Is the Laplacian of $E_\theta(\mathbf{x})$. Now, by adjusting the parameters to minimise the objective function, similar to that seen in Equation (1.3), we have defined an objective function. This process is free from the intractable Z_θ and impractical $\nabla_{\mathbf{x}} \log p_{data}(\mathbf{x})$ partition an energy functions,

$$\mathcal{J}(\theta^*) = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\frac{1}{2} ||\nabla_{\mathbf{x}}\, E_\theta(\mathbf{x})||^2 - \Delta_{\mathbf{x}}\, E_\theta(\mathbf{x}) \right] \tag{A.23}$$

Intuitively, SM can be understood as the means to train EBMs by aligning directions of steepest change (score) between the true data distribution and the models estimate. This method uses a tractable loss that avoids the density estimation and intractable partition function.

A.1.4 Langevin MCMC Method

Consider a statistical system with energy $E_\theta(\mathbf{x})$. From Boltzmann's Law, physical systems naturally evolve toward equilibrium distributions, hence the PDF is defined by the relationship, $p_\theta(\mathbf{x}) \propto e^{-E_\theta(\mathbf{x})}$, which describes physically how particles in thermal equilibrium explore the system. This relationship is aptly named the Boltzmann distribution. Since physical systems follow complex stochastic evolution laws (Bösch et al. 2025), $E_\theta(\mathbf{x})$ can be dynamically estimated through the Langevin equation, allowing sampling.
For sampling from an EBM, Langevin dynamics can be applied to simulate the stochastic process, whose equilibrium distribution matches the target distribution. This process is described by the over-damped Langevin stochastic differential equation (SDE),

$$d\mathbf{x}_t = -\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t)\, dt + \sqrt{2}\, d\mathbf{W}_t \tag{A.24}$$

where $d\mathbf{W}_t$ denotes Brownian motion. The drift term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$ directs the dynamics toward regions of lower energy, while the noise term ensures stochastic exploration of the state space (Parisi 1981).
To approximate this continuous process, we apply the Euler-Maruyama discretisation with step size ϵ, yielding the Unadjusted Langevin Algorithm (ULA). This results in the stationary distribution $p(\mathbf{x}) \propto e^{-E_\theta(\mathbf{x})}$ being modelled via (Song & Ermon 2020),

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\epsilon}{2} \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_{t-1}) + \sqrt{\epsilon}\, \mathbf{z}_t \tag{A.25}$$

where $\mathbf{z}_t \sim \mathcal{N}(0, I)$ is Gaussian noise. This iterative scheme produces approximate samples from $p_\theta(\mathbf{x})$, enabling efficient sampling from EBMs without requiring evaluation of the partition function.

A.1.5 Feynman-Kac Reverse Process Derivation

Recalling Equation (4.6), the time inhomogeneous energy is defined by,

$$E(x,t) = -\log \mathbb{E}_{X_s \sim Q}\left[f(X_T) \exp\left(-\int_t^T V(X_s, s)\, ds \right) + \int_t^T g(X_\tau, \tau) \exp\left(-\int_t^\tau V(X_r, r)\, dr \right) d\tau \right]$$

Allowing source time $\tau = s_j$, and discretising over K time steps, we simplify the expression into components,
To simplify the expression, define

$$\gamma^{(i)} = f(X_T^{(i)}) \exp\left(-\int_t^T V(X_s^{(i)}, s)\, ds \right) \approx f(X_T^{(i)}) \exp\left(-\sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\, \Delta s \right), \quad \text{(A.26)}$$

and

$$\beta^{(i)} = \int_t^T g(X_\tau^{(i)}, \tau) \exp\left(-\int_t^\tau V(X_r^{(i)}, r)\, dr \right) d\tau \approx \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \exp\left(-\sum_{r=0}^{j-1} V(X_{s_r}^{(i)}, s_r)\, \Delta s \right) \Delta s. \quad \text{(A.27)}$$

This gives the simplified expression for the time inhomogeneous energy function as,

$$E(x,t) = -\log \mathbb{E}_{X_s \sim Q}\left[\gamma^{(i)} + \beta^{(i)} \middle| X_t = x \right]$$

Applying the Monte Carlo potential estimator (MCPE),

$$\hat{E}(x,t) = -\log\left[\frac{1}{M} \sum_{i=1}^{M} \left(\gamma^{(i)} + \beta^{(i)} \right) \right] \quad \text{(A.28)}$$

And further re-expression of terms,

$$\alpha^{(i)} = \gamma^{(i)} + \beta^{(i)} \quad \text{(A.29)}$$

so that the MCPE is expressed by,

$$\hat{E}(x^{(i)}, t^{(i)}) = -\log\left[\frac{1}{M} \sum_{i=1}^{M} \left(\alpha^{(i)} \right) \right] \quad \text{(A.30)}$$

Then allow the MC partition function to be expressed as,

$$\hat{Z}(x,t) = \frac{1}{M} \sum_{i=1}^{M} \alpha^{(i)} \quad \Longrightarrow \quad \hat{E}(x,t) = -\log \hat{Z}(x,t) \quad \text{(A.31)}$$

Simply applying the chain rule gives,

$$\nabla_x \hat{E}(x,t) = -\frac{\nabla_x \hat{Z}(x,t)}{\hat{Z}(x,t)} \quad \text{(A.32)}$$

Now, analysing the derivative of the partition,

$$\nabla_x \hat{Z}(x,t) = \frac{1}{M} \sum_{i=1}^{M} \nabla_x \alpha^{(i)} \tag{A.33}$$

Leading to the derivative of the MCPE expression,

$$\nabla_x \hat{E}(x,t) - \frac{\sum_{i=1}^{M} \nabla_x \alpha^{(i)}}{\sum_{i=1}^{M} \alpha^{(i)}} \tag{A.34}$$

Applying log laws allows the derivative of the composite terminal and source function, $\alpha^{(k,i)}$ to be expressed as,

$$\nabla_x \alpha^{(i)} = \alpha^{(i)} \nabla_x \log \alpha^{(i)} \tag{A.35}$$

This gives the MCPE derivative expression,

$$\nabla_x \hat{E}(x,t) = -\frac{\sum_{i=1}^{M} \alpha^{(i)} \nabla_x \log \alpha^{(i)}}{\sum_{i=1}^{M} \alpha^{(i)}} \tag{A.36}$$

Simplifying into the form,

$$\nabla_x \hat{E}(x,t) = -\sum_{i=1}^{M} \hat{\alpha}^{(i)} \nabla_x \log \alpha^{(i)} \tag{A.37}$$

where,

$$\hat{\alpha}^{(i)} = \frac{\alpha^{(i)}}{\sum_{i=1}^{M} \alpha^{(i)}} \tag{A.38}$$

Deriving analytical solution to $\nabla_{x^{(k)}} \hat{E}(x,t)$ Recall the composite terminal and source expression,

$$\alpha^{(i)} = \gamma^{(i)} + \beta^{(i)} \tag{A.39}$$

Expanding terms gives to their original form gives,

$$\alpha^{(i)} = f(X_T^{(i)}) \exp\left(- \sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\, \Delta s \right) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \exp\left(- \sum_{r=0}^{j-1} V(X_{s_r}^{(i)}, s_r)\, \Delta s \right) \Delta s \tag{A.40}$$

Now factorising the expression,

$$\alpha^{(i)} = \exp\left(- \sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\, \Delta s \right) \cdot \left(f(X_T) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \Delta s \right) \tag{A.41}$$

To simplify the notation allow,

$$\phi^{(i)} = -\sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\,\Delta s \tag{A.42}$$

To evaluate the derivative of $\alpha^{(i)}$, which is essential for evaluating the expression $\nabla_x \log \alpha^{(i)} = \frac{\nabla_x \alpha^{(i)}}{\alpha^{(i)}}$, recognise,

$$\nabla_x\, \alpha^{(i)} = \nabla_x \left[\exp(\phi^{(i)}) \left(f(X_T) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\Delta s \right) \right] \tag{A.43}$$

Applying the product rule,

$$\nabla_x\, \alpha^{(i)} = \nabla_x \exp(\phi^{(i)}) \cdot \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\Delta s \right) + \exp(\phi^{(i)}) \cdot \nabla_x \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\Delta s \right) \tag{A.44}$$

Allowing the separation into separate components. Taking the expression,

$$\exp(\phi^{(i)}) \cdot \nabla_x \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\Delta s \right) \tag{A.45}$$

we recognise,

$$\nabla_x f(X_T^{(i)}) + \sum_{j=0}^{K-1} \nabla_x\, g(X_{s_j}^{(i)}, s_j)\Delta s \tag{A.46}$$

Which allows the application of the chain rule to each variable,

$$\nabla_x\, f(X_T^{(i)}) = \frac{\partial f}{\partial X_T^{(i)}} \frac{\partial X_T^{(i)}}{\partial x} \qquad \nabla_x\, g(X_{s_j}^{(i)}, s_j) = \frac{\partial g}{\partial X_{s_j}^{(i)}} \frac{\partial X_{s_j}^{(i)}}{\partial x} \tag{A.47}$$

As well as evaluation of the exponent term,

$$\nabla_x \exp(\phi^{(i)}) = \exp(\phi^{(i)}) \;\cdot\; \nabla_x \phi^{(i)} \tag{A.48}$$

By expanding the exponent back to its negative sum potential form,

$$\nabla_x \phi^{(i)} = -\sum_{j=0}^{K-1} \nabla_x V(X_{s_j}^{(i)}, s_j)\,\Delta s \tag{A.49}$$

Allowing the application of the chain rule once again,

$$\nabla_x \phi^{(i)} = -\sum_{j=0}^{K-1} \frac{\partial V}{\partial X_{s_j}^{(i)}} \cdot \frac{\partial X_{s_j}^{(i)}}{\partial x}\,\Delta s \tag{A.50}$$

To solve the derivative, $\frac{\partial X_{s_j}^{(i)}}{\partial x}$, recall the diffusion process defines a drift for each step in X_{s_j},

$$X_{s_j+1}^{(i)} = X_{s_j}^{(i)} e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s} - (1 - e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s})\mu + \sqrt{\sigma^2(1 - e^{-\Sigma^{-1}\Delta\Lambda_s})} \cdot Z_t \tag{A.51}$$

For Brownian motion with drift we can assume that the point $X_{s_j}^{(i)}$ can be represented by the initial position X_{s_0}, plus the sum of the drift and noise terms,

$$X_{s_j}^{(i)} = X_{s_0}^{(i)} \prod_{\iota=1} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} + \sum_{k=0}^{j-1} \left[(1 - e^{-\frac{1}{2}\Sigma^{-1}\Delta\Lambda_s})\mu + \sqrt{\sigma^2(1 - e^{-\Sigma^{-1}\Delta\Lambda_s})} \cdot Z_t \right] \tag{A.52}$$

Which avoids the error in drift accumulation, directly sampling the transition. Therefore the partial derivative w.r.t. X_{s_0}

$$\frac{\partial X_{s_j}^{(i)}}{\partial X_{s_0}^{(i)}} = \frac{\partial}{\partial X_{s_0}^{(i)}} \left[X_{s_0}^{(i)} \prod_{i=1}^{n} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} \right] + 0 = I_d \prod_{i=1}^{n} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s} \tag{A.53}$$

Allowing $e^{-\theta_{s_j}} = \prod_{i=1}^{n} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_s}$ denote the attenuation term for path convergence. A similar process can be determined for the terminal and source function derivatives applies to all derivates where some X w.r.t. x, leading to expressions,

$$-\sum_{i=1}^{M} \frac{\partial V_{s_j}^{(i)}}{\partial X_{s_j}^{(i)}} \frac{\partial X_{s_j}^{(i)}}{\partial x} \Delta s = -\sum_{i=1}^{M} \frac{\partial V_{s_j}^{i}}{\partial X_{s_j}} I_d \cdot e^{-\theta_{s_j}} \Delta s \tag{A.54}$$

$$\frac{\partial f}{\partial X_T^{(i)}} \frac{\partial X_T^{(i)}}{\partial x} = \frac{\partial f}{\partial X_t^{(i)}} I_d \cdot e^{-\theta_T} \tag{A.55}$$

Where $e^{-\theta_T} = \prod_{i=1}^{n} e^{-\frac{1}{2}\Sigma^{-1}\Lambda_T}$ describes the terminal attenuation rate for which $\Lambda_T = \int_0^T \beta_s \Delta s$, is the cumulative product across the entire noise schedule.

$$\frac{\partial g}{\partial X_{s_j}^{(i)}} \frac{\partial X_{s_j}^{(i)}}{\partial x} = \frac{\partial g}{\partial X_{s_j}^{(i)}} I_d \cdot e^{-\theta_{s_j}} \tag{A.56}$$

Since the potential energy $V(X_s, s)$ is a known analytical solution, the score can be evaluated at point X_{s_j},

$$\nabla_x V(X_{s_j}^{(i)}) = \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)}) \tag{A.57}$$

Now deriving the full $\hat{E}(x^{(i)}, t^{(i)})$ for path i, the expression requires the full expansion of $\nabla_x \log \alpha^{(i)}$ expression by substituting in the known analytical solutions. Recall,

$$\nabla_{X_{s_j}^{(i)}} \alpha^{(i)} = \nabla_{X_{s_j}^{(i)}} e^{\phi^{(i)}} \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \Delta s \right) + e^{\phi^{(i)}} \nabla_{X_{s_j}^{(i)}} \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \Delta s \right) \tag{A.58}$$

Now fully expressing each term using the expressions above,

$$\nabla_{X_{s_j}^{(i)}} e^{\phi^{(i)}} = e^{\phi^{(i)}} \nabla_{X_{s_j}^{(i)}} \phi^{(i)} \tag{A.59}$$

Expanding terms,

$$e^{\phi^{(i)}} \nabla_x \phi^{(i)} = -\exp\left(-\sum_{j=0}^{K-1} V(X_{s_j}^{(i)}, s_j)\, \Delta s \right) \cdot \sum_{i=1}^{M} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)}, s_j)\, \Delta s \tag{A.60}$$

Focusing on the sum derivative term,

$$-\sum_{i=1}^{M} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)})\, \Delta s = -\sum_{i=1}^{M} \frac{\partial V_{s_j}^{(i)}}{\partial X_{s_j}^{(i)}}\, I_d \cdot e^{-\theta_{s_j}}\, \Delta s \tag{A.61}$$

$$-\sum_{i=1}^{M} \frac{\partial V}{\partial X_{s_j}^{(i)}}\, I_d \cdot e^{-\theta_{s_j}}\, \Delta s = -\sum_{i=1}^{M} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)}) \cdot e^{-\theta_{s_j}}\, \Delta s \tag{A.62}$$

Now evaluating the second component of the expression,

$$\nabla_{X_{s_j}^{(i)}} \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \Delta s \right) \tag{A.63}$$

Substituting in our known expressions for the terminal and source variable derivatives,

$$\frac{\partial f}{\partial X_T^{(i)}} \mathbb{I}_d \cdot e^{-\theta_{s_j}} + \sum_{j=0}^{K-1} \frac{\partial g}{\partial X_{s_j}^{(i)}} I_d \cdot e^{\theta_{s_j}^{(i)}}\, \Delta s \tag{A.64}$$

Combining all back into $\nabla_x \alpha^{(i)}$ expression,

$$\nabla_x \alpha^{(i)} = -\sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)})\, e^{-\phi^{(i)}}\, \Delta s \cdot \left(f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j) \Delta s \right) + e^{\phi^{(i)}} \left(\frac{\partial f}{\partial X_T^{(i)}} e^{-\theta_T} + \sum_{j=0}^{K-1} \frac{\partial g}{\partial X_{s_j}^{(i)}} e^{-\theta_{s_j}}\, \Delta s \right) \tag{A.65}$$

By substituting in derived terms for derivative functions we arrive at the expression,

$$\nabla_{X_{s_j}^{(i)}} \log \alpha^{(i)} = -\left(\sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)}) \cdot e^{-\theta_{s_j}}\, \Delta s \right) + \frac{\nabla_{X_T^{(i)}} f(X_T^{(i)})\, e^{-\theta_T} + \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} g(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s}{f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\, \Delta s} \tag{A.66}$$

As long as functions $V(X_{s_j}, s_j)$, $f(X_T)$ and $g(X_{s_j}, s_j)$ have differentiable analytical solutions, the expression retains a fully closed form derivation. This implies since the free energy is a known closed-form function of the data, and all paths are differentiable w.r.t. their initial conditions, the gradient can be expressed in terms of the sampled trajectory time steps. The final full expression for the MCPE gradient, $\nabla_x \hat{E}(x^{(i)}, t^{(i)})$,

$$\nabla_x \hat{E}(x^{(i)}, t^{(i)}) = -\sum_{i=1}^{M} \hat{\alpha}^{(i)} \left(\sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} V(X_{s_j}^{(i)})\, e^{-\theta s_j}\, \Delta s \right) + \psi^{(i)} \tag{A.67}$$

where,

$$\psi^{(i)} := \frac{\nabla_{X_T^{(i)}} f(X_T^{(i)})\, e^{-\theta_T} + \sum_{j=0}^{K-1} \nabla_{X_{s_j}^{(i)}} g(X_{s_j}^{(i)}, s_j)\, e^{-\theta_{s_j}}\, \Delta s}{f(X_T^{(i)}) + \sum_{j=0}^{K-1} g(X_{s_j}^{(i)}, s_j)\, \Delta s}$$