UNIVERSITY OF CAMBRIDGE
Department of Engineering

# Learning Gradient Variance Control with Thermodynamic Integration: Applications in Deep Generative Image Modelling

Project Code: **D-mag92-1**

Author Name: Prithvi Raj

Supervisor: Prof. Mark Girolami

Date: 26/04/2024

I hereby declare that, except where specifically indicated, the work submitted herin is my own original work.

*Signed* Prithvi Raj *date* 26/04/2024

# Learning Gradient Variance Control with Thermodynamic Integration: Applications in Deep Generative Image Modelling

**Prithvi Raj**
Department of Engineering
University of Cambridge
pr478@cam.ac.uk

**Mark Girolami**
Department of Engineering
University of Cambridge
mag92@cam.ac.uk

**Abstract**

The performances of deep generative models depend on the distributional characteristics of their learning gradients. Despite this, the exact influence of learning gradient variance remains poorly understood, and investigations into the topic are bounded by our limited ability to control gradient variance. For example, managing gradient variance through batching alone is challenging, especially under constrained computational resources.

To address this, we propose leveraging Thermodynamic Integration as a means of robustly controlling the learning gradient variance. This is achieved by parameterising the temperature schedule used to evaluate the thermodynamic integral. This parameterisation allows us to exert precise control over the variances in estimates of latent space variables derived through Markov chain Monte Carlo (MCMC) sampling, as well as the error in Monte Carlo estimates of the mean.

The method is subsequently proven and applied to investigate the relationship between learning gradient variance and the fidelity of images generated by the latent space energy-based prior model introduced by Pang et al. [16]. This study reveals that although there is a notable relationship between learning gradient variance and image fidelity, learning gradient variance alone is inadequate as a predictor of the generative capacity of the latent space energy-based prior model.

Instead, the study demonstrates that the temperature schedule itself exerts an even greater influence on image fidelity, serving as a direct reflection of the balance between exploration and exploitation that the deep generative model maintains over the loss landscape.

## Contents

# 1 Introduction

## 1.1 The state of deep generative models in engineering

Deep generative models belong to a category of machine learning algorithms characterised as neural networks for generating new data samples, such as images or text. Generative image models are especially poised to revolutionise various engineering domains, particularly in fields such as satellite imagery analysis, computer-aided design, and medical imaging.

For example, a notable case study is presented in [20], where researchers tackle the challenge of limited annotated satellite imagery data. They achieve this by using generative image models to expand datasets with new, artificial images, thereby facilitating important downstream tasks like semantic segmentation - a tool used to support engineers with the identification and characterisation of problems such as road defects, wildfires, or deforestation.

Beyond its use in data augmentation, another application which generative image modelling has shown promise for is topology optimisation, as demonstrated in prior studies like [10], [17], and [14]. Specifically, neural networks can be employed to craft optimal physical and material structures that maximise mechanical performance within predefined design parameters for coupled multiphysics problems.

The use of generative image models enables engineers to navigate intricate design spaces, uncovering optimal configurations that traditional methods may disregard. This automated approach holds potential across various sectors, from aerospace to biomedical, facilitating the discovery of innovative solutions in a flexible, non-iterative, and more efficient manner.

Fig. 1 showcases an example of topology optimisation for structural engineering from [14]. Ground truth structures were generated using the Solid Isotropic Material with Penalisation (SIMP) solver, a gradient-based, iterative Finite Element Analysis (FEA) method.

Each 2D image depicts a physically feasible structure, accompanied by its compliance error relative to the ground truth. The generative models were trained on 2D images corresponding to optimal structures for a diverse range of input conditions, encompassing various physical fields, boundary conditions, and volume fractions.

The examples are generated for randomly selected samples from both the "level 1" and "level 2" test datasets, each posing their unique challenges regarding generalisability.

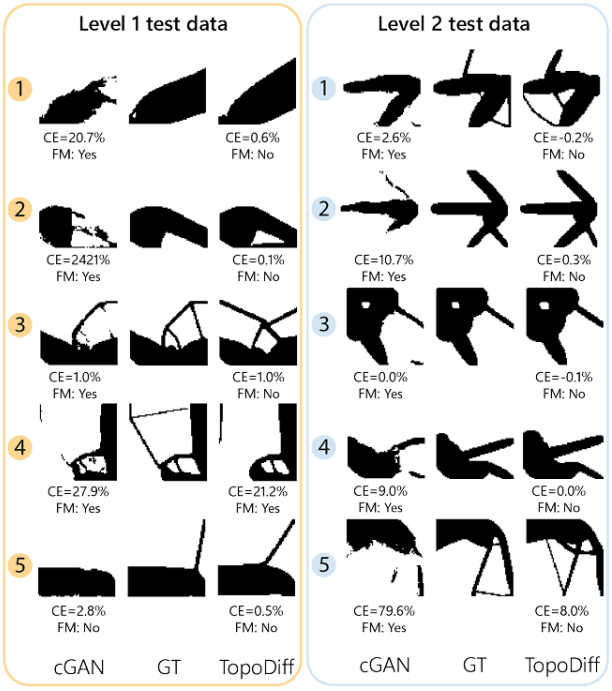



Figure 1: Structures obtained via a ground truth (GT) method alongside those generated by a conditional generative adversarial network (cGAN) model and a conditional diffusion model (TopoDiff) from the study in [14].

The level 1 test set comprised novel combinations of constraints seen during training, with unseen configurations of the input conditions. Conversely, the level 2 test set presented entirely new combinations of constraints, including out-of-distribution boundary conditions, i.e. conditions absent from the training data altogether.

Therefore, level 1 testing assessed the model's performance on previously unseen combinations of conditions encountered during training, whereas level 2 testing gauged its generalisation capability to entirely novel boundary conditions. The latter, represented a more challenging out-of-distribution test set, serving to evaluate the model's ability to generalise beyond its training data.

In both cases, the study showed that the FEA-based method outperformed the neural networks in terms of compliance error. However, its iterative nature renders it inefficient, and advancements in machine learning are narrowing the performance gap.

Overall, for image models to outperform traditional iterative solvers like SIMP, they must first improve their generalisation capacities and adeptness in navigating probabilistic landscapes. Achieving this requires embracing a distributional perspective in their learning processes, which in turn demands significant research endeavors focused on advancing generative neural networks.

### 1.2 The learning gradient distribution

Deep generative models undergo training primarily by minimising a loss function, denoted as $\mathcal{L}(\theta, \mathbf{x})$, achieved through iterative adjustments of the neural network parameters, represented as $\theta$. The form of these adjustments are evaluated through gradient-based optimisation techniques, such as stochastic gradient descent (SGD) [18] and Adam optimisation [11].

Importantly, these methods require the learning gradient, $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$, i.e. the gradient of the loss with respect to the parameters, evaluated at a training sample $\mathbf{x}$. However, due to the probabilistic nature of the data and the inherent noise introduced by gradient-based optimisation methods, the learning gradient forms a non-deterministic probability distribution.

In engineering applications such as topology optimisation, characterised by inherently noisy and complex data, this probability distribution can become exceedingly intricate. This complexity poses significant challenges, necessitating robust strategies to effectively leverage and navigate any probabilistic information within the learning process.

In agreement with the previous work of [5], we therefore argue that adopting a distributional perspective on gradient-based optimisation and exploring the characteristics of $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$ are crucial steps toward enhancing optimisation speed and the model's ability to generalise beyond the training dataset and produce high fidelity, diverse samples.

However, exerting explicit control over the shape of this distribution presents a persistent obstacle that has prevented researchers from fully grasping its exact influence in the optimisation of deep neural networks. Hence, we demonstrate Thermodynamic Integration as a method to effectively shape this distribution by directly parameterising its variance, $\mathrm{Var}_\theta \left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$, aiming for its adoption in similar investigations concerning gradient-based learning.

The study was implemented in JAX at `https://github.com/PritRaj1/JAX-ThermoEBM`. The raw experimental readings have also been provided as part of the codebase.

### 1.3 Previous studies

1. A study showcased in [2] demonstrated that employing larger mini-batch sizes, resulting in smaller gradient noise during Generative Adversarial Network (GAN) training, substantially enhanced the quality of generated samples and improved training stability.

- This suggests that smaller variances in $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$ are favourable, which is generally not consistent across different studies.
- This may also suggest that batch size emerges as a promising method to control the learning gradient variance. However, we later discuss and investigate why this is not the case in Sections 2.3.1 and 5.1.1.

2. Contrastingly, the previous work of [15] revealed that injecting additional gradient noise into very deep generative networks enhanced model performance by addressing overfitting issues and, notably, reducing training loss.
   - This instead suggests that larger variances in $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$ are preferable in some cases.
   - However, despite being easy to implement, injecting noise is not a robust and controllable means of shaping the distribution of $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$.
3. The study in [5] presents Gradient Clustering (GC) as a method to minimise gradient variance.
   - However, the work does not conclusively show that minimising gradient noise via GC consistently leads to faster convergence or improved accuracy.
   - It also does not present Gradient Clustering as a means of arbitrarily controlling gradient variance, which is an advantage that Thermodynamic Integration presents in investigative research applications. Minimising variance might be justifiable if $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$ were a value, but it is instead a distribution.

Collectively, these prior studies demonstrate the limited comprehension among machine learning practitioners regarding the distributional characteristics of $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$, despite its pivotal role in optimising the performance of deep generative models.

## 2 Theory

### 2.1 Introductory concepts

This section provides an overview of essential concepts in generative modeling for images. It introduces the fundamental principles of latent space modeling and Markov chain Monte Carlo (MCMC) sampling techniques.

#### 2.1.1 Latent space generative modelling

Latent space modeling is a ubiquitous concept in the realm of deep generative modelling for image generation. It involves the creation of a compact, low-dimensional representation of a dataset that effectively captures its essential features and underlying structure.

Subsequently sampling from the latent representation and transforming back to the data space allows for the generation of new images, providing a powerful, scalable, and controllable way to model and generate complex image data using deep neural networks.

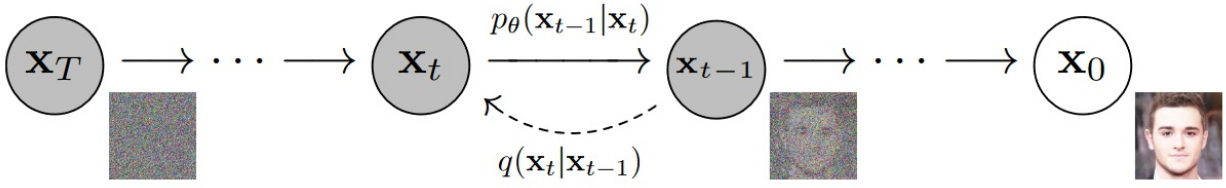



Figure 2: Example of latent-space learning: Denoising Diffusion Model (DDM) [8].

Presented above is an illustration of the Denoising Diffusion Model (DDM), included here to help introduce latent space modelling. A Markov chain process is used to gradually inject noise to provide a mapping between observable, yet intractable data spaces and simpler latent representations that are tractable through various sampling techniques. The noise-additive diffusion process is parameterised and learned using variational inference.

The Markov chain transitions are then reversed to enable seamless image generation after first sampling from the latent distribution. Other exemplar latent space generative models include variational autoencoders (VAEs), generative adversarial networks (GANs), and flow-based models.

### 2.1.2 Markov chain Monte Carlo (MCMC) methods

Another universal concept in contemporary machine learning literature is Markov chain Monte Carlo (MCMC) sampling as a means of estimating draws from complex, intractable probability distributions. In the context of latent space modelling, MCMC sampling serves as a fundamental technique for learning and generating samples from latent prior or posterior distributions, which are usually too highly-defined to generate through standard random number generators available through typical computing and software libraries.

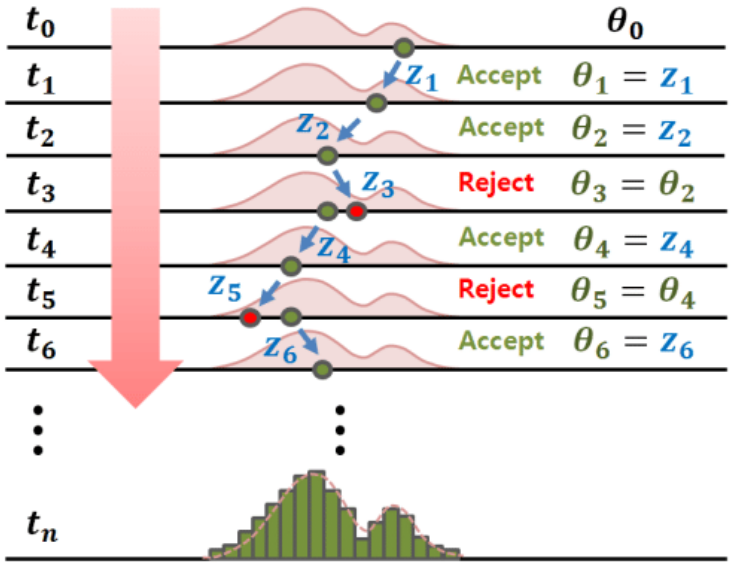



Figure 3: An illustration of the Markov chain Monte Carlo (MCMC) sampling algorithm [9]. In this particular example, an acceptance criterion is included.

A Markov chain is constructed such that its stationary distribution is the target probability distribution that the sample is to be drawn from.

The process starts from an initial state chosen arbitrarily or according to some predefined distribution. Then, at each step of the Markov chain, a proposal state is generated from a transition operator based on the current state.

The proposal is then either accepted or rejected based on an acceptance probability, determined by comparing the probability density of the proposed state under the target distribution with that of the current state. If the proposed state is more probable, it is accepted; otherwise, it may be accepted with a probability proportional to the ratio of the densities.

These accepted transitions and rejections are repeated iteratively. Over many iterations, the Markov chain explores the state space, gradually converging to the stationary distribution. Specifically in Fig. 3, $\theta_i$ denotes the state at iteration $i$, $z_i$ represents the proposed transition, and $t_i$ indicates the time corresponding to iteration $i$.

### 2.2 The latent space energy-based prior model

This Section outlines the specific deep generative model under scrutiny, and details sources of error inherent in its Monte Carlo methods. These sources contribute to variance in the marginal likelihood and consequently affect the learning gradient noise.

#### 2.2.1 Introducing the model

The latent space energy-based prior model proposed in [16] leverages both latent space representations and Markov chain Monte Carlo (MCMC) techniques to generate high-fidelity images, despite tolerating a notable amount of statistical flexibility in its MCMC sampling procedure.

This statistical flexibility introduces a degree of variance into the latent variable estimates produced by MCMC sampling, as well as its overall marginal likelihood evaluation. This marks the latent space energy-based prior model as a prime candidate for investigating the influence of learning gradient variance using Thermodynamic Integration.

#### 2.2.2 Defining the networks

The novelty of this model was its inclusion of an energy-based model (EBM) to learn the latent representation. After first sampling the latent variable $\mathbf{z}$ from a simple prior distribution, e.g. $\pi_0(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \sigma_0\mathbf{I})$, (which can be realised easily using standard random number generators), the sample was then passed through an EBM to exponentially-tilt the latent distribution producing a parameterised latent prior:

$$p_\alpha(\mathbf{z}) = \frac{1}{Z_\alpha} \exp(f_\alpha(\mathbf{z})) \cdot \pi_0(\mathbf{z}) \tag{1}$$

Here, $\alpha$ represents the EBM's learnable parameters, $f_\alpha(\mathbf{z})$ denotes the output of the EBM network when fed an input $\mathbf{z} \sim \pi_0(\mathbf{z})$, and $Z_\alpha$ represents the normalisation constant included to ensure the validity of the probability distribution.

A sample from this parametrised prior, $p_\alpha(\mathbf{z})$, subsequently serves as an input to the generator network (GEN), which maps the latent representation back to the data space:

$$\tilde{\mathbf{x}} = g_\beta(\mathbf{z}) + \epsilon \tag{2}$$

Here, $\beta$ denotes the learnable parameters of the GEN, $\tilde{\mathbf{x}}$ signifies the generated data, $g_\beta(\mathbf{z})$ represents the output of the GEN, and $\epsilon \sim \mathcal{N}(0, \sigma_l\mathbf{I})$ indicates the standard noise term of the top-down network of the VAE class of generative models, (which is the family of networks that the GEN can be characterised into).

The effective parameterisation of the latent prior distribution in Eq. 1 enables the iterative refinement of the latent representation based on observations from the dataset. This is essentially achieved through training the EBM network to shape the prior into a distribution that loosely satisfies $p_\alpha(\mathbf{z}) \approx p_\theta(\mathbf{z}|\mathbf{x})$, where $\theta$ is used here to define the set containing all model parameters, i.e. $\alpha, \beta \in \theta$.

The posterior distribution, $p_\theta(\mathbf{z}|\mathbf{x})$, can be considered a data-informed latent distribution. It is a distribution of $\mathbf{z}$ conditioned on observations from the real dataset used to train the model, $\mathbf{x}$. Therefore, the prior model in 1 can be interpreted as an energy-based, data-informed correction or exponential tilting of the original prior distribution $\pi_0(\mathbf{z})$, informed by features of $\mathbf{x}$. This allows the EBM to exert control over the latent distribution and shape it accordingly, resulting in a more refined latent representation that is better suited to capturing essential features in the latent space.

When creating a new generation of $\tilde{\mathbf{x}}$, the latent variable $\mathbf{z}$ is then sampled directly from $p_\alpha(\mathbf{z})$ and passed through Eq. 2. It is expected that after training the EBM, $p_\alpha(\mathbf{z})$ is similar enough to $p_\theta(\mathbf{z}|\mathbf{x})$ that the process is akin to sampling from $p_\theta(\mathbf{z}|\mathbf{x})$ itself and passing the result through Eq. 2, which is not feasible without having new reference data $\mathbf{x}$, required to evaluate samples from $p_\theta(\mathbf{z}|\mathbf{x})$.

### 2.2.3 Training the networks

The complete model, comprising both networks, is then trained using the maximum likelihood approach:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p_\theta(\mathbf{x}) = \underset{\theta}{\operatorname{argmin}}\ -p_\theta(\mathbf{x}) = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}(\theta, \mathbf{x}) \tag{3}$$

Here, $\mathcal{L}(\theta, \mathbf{x})$ denotes the resulting loss function used to train the model.

Training the two neural networks involves updating the networks' parameters in the direction of the gradient of the loss function with respect to those parameters:

$$\nabla_\theta \mathcal{L}(\theta, \mathbf{x}) = -\nabla_\theta \log(p_\theta(\mathbf{x})) \tag{4}$$

This can be separated into two terms corresponding to each set of model parameters, by first rewriting the likelihood as a marginalisation over the entire latent posterior distribution:

$$\begin{aligned} \nabla_\theta \log(p_\theta(\mathbf{x})) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log(p_\theta(\mathbf{x}, \mathbf{z}))\right] \\ &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log(p_\alpha(\mathbf{z}) p_\beta(\mathbf{x}|\mathbf{z}))\right] \\ &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\alpha \log(p_\alpha(\mathbf{z})) + \nabla_\beta \log(p_\beta(\mathbf{x}|\mathbf{z}))\right] \end{aligned} \tag{5}$$

The term corresponding to the EBM model can be expanded as

$$\nabla_\alpha \log p_\alpha(\mathbf{z}) = \nabla_\alpha f_\alpha(\mathbf{z}) - \mathbb{E}_{p_\alpha(\mathbf{z})}\left[\nabla_\alpha f_\alpha(\mathbf{z})\right]$$

With this in mind, the two learning gradients required to update each neural network's set of parameters are therefore:

$$\delta_\alpha(\mathbf{x}) = \nabla_\alpha \log p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\alpha f_\alpha(\mathbf{z})\right] - \mathbb{E}_{p_\alpha(\mathbf{z})}\left[\nabla_\alpha f_\alpha(\mathbf{z})\right] \tag{6}$$

$$\delta_\beta(\mathbf{x}) = \nabla_\beta \log p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\beta \log p_\beta(\mathbf{x}|\mathbf{z})\right] \tag{7}$$

In Eq. 6, $\alpha$ is updated based on the difference between the EBM's output using $\mathbf{z}$ sampled from two different probability distributions, $p_\theta(\mathbf{z}|\mathbf{x})$ and $p_\alpha(\mathbf{z})$. An instance of $\mathbf{z}$ sampled from $p_\theta(\mathbf{z}|\mathbf{x})$ can be considered to be inferred from empirical observations $\mathbf{x}$. An instance of $\mathbf{z}$ sampled from $p_\alpha(\mathbf{z})$ is solely dependent on the EBM's parameters, uninformed by $\mathbf{x}$. Therefore, the EBM's update step serves to tune the exponential-tilting of Eq. 1 to shape $p_\alpha(\mathbf{z})$ into a distribution that is more representative of the fundamental features present in $\mathbf{x}$, despite the distribution's lack of inherent dependence on $\mathbf{x}$.

In the context of image generation, the likelihood term in Eq. 7 is given as:.

$$\log p_\beta(\mathbf{x}|\mathbf{z}) = -\frac{\|\mathbf{x} - g_\beta(\mathbf{z})\|^2}{2\sigma_l^2} + \text{constant} \tag{8}$$

### 2.2.4 Sampling $p_\alpha(\mathbf{z})$ and $p_\theta(\mathbf{z}|\mathbf{x})$ with Langevin dynamics

To compute the expectations in Eq. 6 and 7, MCMC sampling is used to estimate samples of $\mathbf{z}$ from $p_\alpha(\mathbf{z})$ and $p_\theta(\mathbf{z}|\mathbf{x})$, which are otherwise intractable.

In [16], the particular MCMC technique used was unadjusted Langevin Sampling. Given a target distribution, $q(\mathbf{z})$, the MCMC update step is computed as:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + s\nabla_\mathbf{z} \log q(\mathbf{z}_k) + \sqrt{2s}\epsilon_\mathbf{k}, \quad k = 1, \ldots, K_\mathbf{z} \tag{9}$$

Here, $K_\mathbf{z}$ denotes the total number of steps that the MCMC sampling loop is conducted for, $k$ indexes the incremental step of the Langevin algorithm, $s$ is a small step size, and $\epsilon_\mathbf{k} \sim \mathcal{N}(\mathbf{0}, \mathbf{I_d})$ is Gaussian white noise. The starting sample, $\mathbf{z}_0$, is initialised from a predefined distribution, $\mathbf{z}_0 \sim q_0(\mathbf{z})$, and the resulting $\mathbf{z}_{K_\mathbf{z}}$ is the final sample from the estimated target distribution, $q(\mathbf{z})$.

In the case of sampling from $p_\alpha(\mathbf{z})$:

$$q(\mathbf{z}) = p_\alpha(\mathbf{z}), \tag{10}$$
$$q_0(\mathbf{z}) = \pi_0(\mathbf{z}), \tag{11}$$
$$\nabla_\mathbf{z} \log q(\mathbf{z}_k) = \nabla_\mathbf{z} \log p_\alpha(\mathbf{z}) = \nabla_\mathbf{z} f_\alpha(\mathbf{z}) - \frac{\mathbf{z}}{\sigma_0^2} \tag{12}$$

In the case of sampling from $p_\theta(\mathbf{z}|\mathbf{x})$:

$$q(\mathbf{z}) = p_\theta(\mathbf{z}|\mathbf{x}), \tag{13}$$
$$q_0(\mathbf{z}) = \pi_0(\mathbf{z}), \tag{14}$$
$$\nabla_\mathbf{z} \log q(\mathbf{z}_k) = \nabla_\mathbf{z} \log p_\theta(\mathbf{z}|\mathbf{x}) \tag{15}$$
$$= \nabla_\mathbf{z} \log p_\beta(\mathbf{x}|\mathbf{z}) + \nabla_\mathbf{z} \log p_\alpha(\mathbf{z}) \tag{16}$$
$$= -\nabla_\mathbf{z} \frac{\|\mathbf{x} - g_\beta(\mathbf{z})\|^2}{2\sigma_l^2} + \nabla_\mathbf{z} f_\alpha(\mathbf{z}) - \frac{\mathbf{z}}{\sigma_0^2} \tag{17}$$

### 2.2.5 Sources of variance in the MCMC procedure

Guaranteeing full convergence to an intricate target distribution in MCMC sampling typically requires an infinite number of steps with infinitesimal step sizes, which is infeasible. Instead, the Langevin sampling loop used in the previous work of [16] adopts a statistically flexible approach in its implementation. This introduces noticeable, unintended variance into the value that the final sample, ($\mathbf{z}_K$), converges on, in addition to the Monte Carlo error later elucidated in Eq. 19.

Firstly, The total MCMC procedure was conducted for a small number of iterations, e.g. $K_{\mathbf{z}|\mathbf{x}} = 20$ in Eq. 9. This limited iteration count diminishes the procedure's capacity to thoroughly explore the target distribution. Consequently, the resulting final sample may have failed to faithfully represent the target distribution.

Such a scenario would have introduced a notable degree of variance, as the samples are less likely to converge towards any specific mode of the target distribution. Instead, they tend to be spread across the distribution's modes, introducing more variance into the final sample.

Additionally, the Langevin sampling loop operated without error correction, unlike the algorithm discussed in Section 2.1.2. All proposal steps in the Langevin sampling loop presented in Eq. 9 were accepted. The procedure did not include any mechanism to reject samples based off the relative probability that the proposal state takes, (as otherwise implemented in Metropolis-adjusted Langevin sampling). Without any rejection mechanism, there is no means of correcting or reducing errors, which may even accumulate during each iteration. This similarly introduced a notable degree of variance into the value that the final sample adopts.

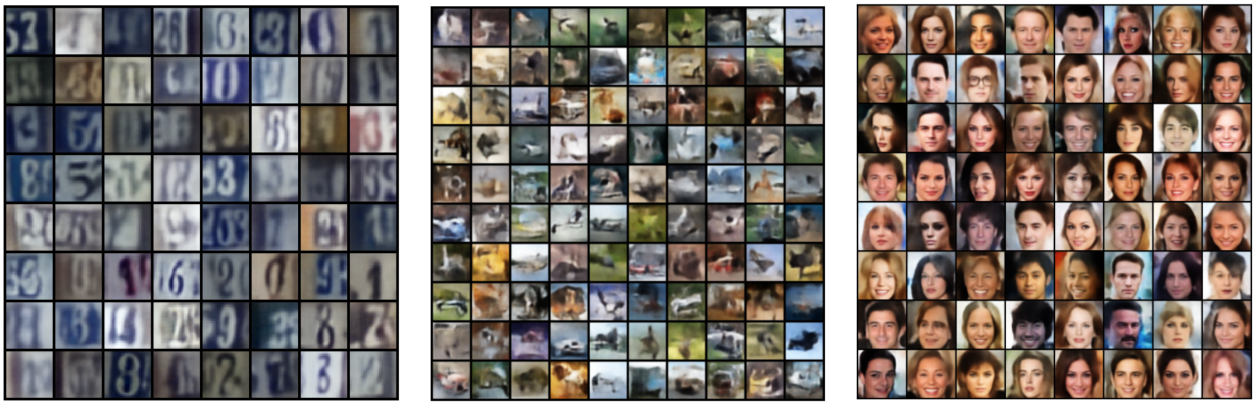
Furthermore, the Langevin step, and learning gradient computations described in Eqs. 6, 7, 12, and 17 require the calculation of expectations. Computing these expectations requires marginalising or integrating over the entire latent space. This is an infeasible computation. Instead, the loss was estimated in [16] through Monte Carlo methods. For example, to approximate the marginal likelihood:

$$p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[p_\theta(\mathbf{x},\mathbf{z})\right] = \int p_\theta(\mathbf{z}|\mathbf{x}) p_\theta(\mathbf{x},\mathbf{z})\, d\mathbf{z} \approx \frac{1}{N}\sum_{i=1}^{N} p_\theta(\mathbf{x},\mathbf{z}^{(i)}) = \overline{\rho}(N) \tag{18}$$

Here, $\mathbf{z}^{(i)}$ represents the $i$th sample taken from the set $\left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N)}\right\}$, with $N$ indicating the total number of samples in the set. However, it can be shown that the Monte Carlo error in this estimator scales with the inverse root of the sample size, $N$ [12]:

$$\sqrt{\mathbb{E}\left[\overline{\rho}(N)^2\right] - \mathbb{E}\left[\overline{\rho}(N)\right]^2} \propto \frac{1}{\sqrt{N}} \tag{19}$$

In the previous work of [16], the sample sizes used in any MCMC estimators were defined through batching and the size of the latent representation. As such, these estimators emerged as significant sources of variance in the learning gradients and MCMC-estimated samples, solely controllable through choosing a sample or batch size. These are not ideal parameters for controlling gradient noise, as discussed in the beginning of Section 2.3. However, despite these sources of variance, Fig. 4 demonstrates the model's capability for generating high fidelity images when trained on the SVHN, CIFAR-10, and CelebA datasets.



(a) SVHN ($32 \times 32 \times 3$) (b) CIFAR10 ($32 \times 32 \times 3$) (c) CelebA ($64 \times 64 \times 3$)

Figure 4: Samples generated by the latent space energy-based prior model, as presented in [16].

This prompts the study's inquiry regarding the significance of Markov chain Monte Carlo (MCMC) methods within latent space modeling for image generation. Is there a correlation between the model's capacity to produce lower variance MCMC estimates and its generative performance? Alternatively, does maintaining a high variance in samples serve a crucial role in sustaining sufficiently large gradient noise, thus aiding in the exploration of the image landscape? Thermodynamic

Integration emerges as a well-suited means of creating the range of variances required to answer these questions, without requiring additional computational resources.

### 2.3 Thermodynamic Integration

This section introduces Thermodynamic Integration, demonstrating how varying temperature schedules influence the variance of the learning gradient and the interplay between the prior and the true Bayesian posterior distributions in posterior sampling.

#### 2.3.1 The case for Thermodynamic Integration

As previously mentioned, trying to reduce the MCMC estimator's variance through batch or sample size alone proves challenging. Firstly, increasing sample size to reduce MCMC error is limited by computational resources. Secondly, for the same number of training examples, a different batch size results in a different frequency of parameter updates per epoch. This increases the difficulty in conducting studies into gradient noise, since parameter updates per epoch is no longer a control variable in the comparison between different models. A similar argument can be made for the size of the latent representation, a crucial parameter that may require consistency between experiments.

Instead, we apply Thermodynamic Integration towards obtaining the marginal likelihood as presented in [3]. For the same sample size, Thermodynamic Integration can be structured to parameterise the learning gradient variance, and achieve its explicit control without requiring more storage. Instead, its implementation requires more compute time, which is influenced by $N_t$ in Eq. 23.

#### 2.3.2 The thermodynamic integral

In place of the formulation and approximation outlined in eqs. 5 and 18, this method exploits the following representation of the logarithm of the marginal likelihood:

$$\log p_\theta(\mathbf{x}) = \int_0^1 \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x},t)} \left[\log p_\beta(\mathbf{x}|\mathbf{z})\right] \, dt \tag{20}$$

Here, $p_\theta(\mathbf{z}|\mathbf{x}, t)$ has been introduced as the power posterior, which can be considered a representation of the original posterior distribution, tempered by a temperature parameter, $t$. This is presented in [6] and organised for our purposes as follows:

$$p_\theta(\mathbf{z}|\mathbf{x}, t) = \frac{p_\beta(\mathbf{x}|\mathbf{z})^t p_\alpha(\mathbf{z})}{\mathcal{Z}(\mathbf{x}|t)}, \quad \text{where } \mathcal{Z}(\mathbf{x}|t) = \int p_\beta(\mathbf{x}|\mathbf{z})^t p_\alpha(\mathbf{z}) d\mathbf{z} \tag{21}$$

Given the formulation in Eq. 21, setting $t = 0$ results in the posterior distribution assuming the form of the prior distribution. As $t$ increments, the posterior gradually adopts a shape more closely resembling the true Bayesian posterior distribution. At $t = 1$, the posterior converges to the true Bayesian posterior distribution.

Integrating over the entire temperature range as specified in Eq. 20 ensures that every tempered posterior distribution plays a role in estimating the marginal likelihood. Therefore, the selection of the temperature schedule directly impacts the estimator's evaluation.

The tempered posterior also alters the update step in Eq. 17, given that the likelihood term has been raised to the power of $t$:

$$
\begin{aligned}
\nabla_{\mathbf{z}} \log q(\mathbf{z}_k) &= \nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}|\mathbf{x}, t) \\
&\propto \nabla_{\mathbf{z}} \log p_\beta(\mathbf{x}|\mathbf{z})^t + \nabla_{\mathbf{z}} \log p_\alpha(\mathbf{z}) \\
&\propto -\nabla_{\mathbf{z}} \frac{t \cdot \|\mathbf{x} - g_\beta(\mathbf{z})\|^2}{2\sigma_l^2} + \nabla_{\mathbf{z}} f_\alpha(\mathbf{z}) - \frac{\mathbf{z}}{\sigma_0^2}
\end{aligned}
\tag{22}
$$

Setting $t = 0$ results in the MCMC sampling procedure yielding an estimate from the prior distribution, while increasing $t$ progressively facilitates exploration of the true posterior distribution. Therefore, aside from addressing error in the Monte Carlo estimator, using the thermodynamic integral also allows for more nuanced MCMC exploration of the posterior distribution, which mitigates the variances associated with the unadjusted Langevin algorithm discussed in Section 2.2.5.

### 2.3.3 Discretisation of the thermodynamic integral

The exact calculation of the expectation in Eq. 20 remains practically infeasible, so following the work of [3], the temperature schedule is instead discretised as:

$$
\{t_1, t_2, \dots t_{N_\mathrm{t}}\} \text{ for } t_i \in [0, 1] \tag{23}
$$

Here, $t_i$ denotes the tempering at the $i$th index of the schedule, and $N_\mathrm{t}$ is the number of temperatures. As such, the the evaluation in Eq. 20 is approximated as:

$$
\log p_\theta(\mathbf{x}) = \frac{1}{2} \sum_i \Delta t_i (E_{i-1} + E_i) + \frac{1}{2} \sum_i D_{\mathrm{KL}}(p_{i-1}||p_i) + D_{\mathrm{KL}}(p_i||p_{i-1}) \tag{24}
$$

Where:

$$
\begin{aligned}
\Delta t_i &= t_i - t_{i-1} \\
E_i &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x}, t_i)} \left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \\
D_{\mathrm{KL}}(p_{i-1}||p_i) &= \int p_\theta(\mathbf{z}|\mathbf{x}, t_{i-1}) \frac{p_\theta(\mathbf{z}|\mathbf{x}, t_{i-1})}{p_\theta(\mathbf{z}|\mathbf{x}, t_i)} dz, \quad \text{(the Kullback-Leibler Divergence)}
\end{aligned}
$$

The above approximation is equivalent to using the trapezium rule for numerical integration. Its full derivation is presented in [3], and is computationally realised using Monte Carlo estimates for each $E_i$, which remains unavoidable and serves as a source of error that is once again limited by the memory requirements associated with large sample sizes.

The second source of error arises from the KL divergence term outlined in Eq. 24, an integration that requires approximation. The extent of error in this approximation is primarily bound by the number of partitions incorporated in the temperature schedule, $N_t$. The constraint on $N_t$ stems from computation time rather than memory resources. A higher value of $N_t$ scales the number of evaluations needed to compute Eq. 24, thereby elongating the time required for the marginal likelihood calculation. This is shown in Section 8.3.1 of the Appendix.

Noting that the power posterior in Eq. 21 is Gaussian distributed, the KL divergence term used for Eq. 24 is computationally realised using the following analytic expression from [3]:

$$
D_{\mathrm{KL}}(p_{i-1}||p_i) = \frac{1}{2} \left( \ln \left( \frac{\det(\Sigma_i)}{\det(\Sigma_{i-1})} \right) + \mathrm{tr}(\Sigma_i^{-1} \Sigma_{i-1}) + (\mu_i - \mu_{i-1})^T \Sigma_i^{-1} (\mu_i - \mu_{i-1}) - d \right) \tag{25}
$$

Here, $\Sigma_i$ and $\Sigma_{i-1}$ are the covariance matrices of the distributions $p_i$ and $p_{i-1}$ respectively; $\mu_i$ and $\mu_{i-1}$ are the means, and $d$ is the dimensionality of the distributions, representing the number of dimensions in the multivariate space.

### 2.3.4 Temperature scheduling

The previous work of [3] also presents an approach to minimise the variance associated with the Monte Carlo estimate of $\log p_\theta(\mathbf{x})$ by optimising the particular choice of temperature schedule.

In this study, we opt to instead parameterise the schedule using a temperature power term, denoted as $p$:

$$t_i = \left(\frac{i}{N_\mathrm{t}}\right)^p \quad \forall i \in \{1, \ldots, N_\mathrm{t}\} \tag{26}$$

Fig. 5 shows the impact of adjusting $p$. A higher $p$ amplifies the influence of posteriors tempered by lower $t$ values, indicating a greater emphasis on the simpler prior distribution and reduced exploration of the true posterior distribution.

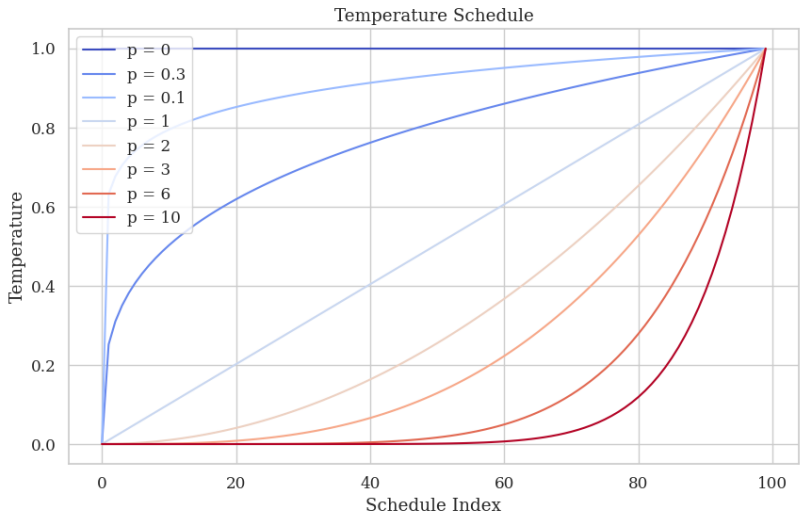



Figure 5: Temperature schedules parameterised by $p$ in Eq. 26.

A temperature schedule with partitions favouring lower temperatures, i.e., $p > 1$, is anticipated to elevate global exploration of $\log p_\theta(\mathbf{x})$, when compared against a temperature schedule characterised by partitions favouring higher temperatures i.e., $0 < p \leq 1$, which is expected to elevate exploitation of its local form.

This is due to the formulation of Eq. 21. Lower temperatures favour more exploration of $p_\alpha(\mathbf{z})$, a simpler distribution compared to the true Bayesian posterior distribution, $p_\theta(\mathbf{z}|\mathbf{x})$, which higher temperatures favour the exploration of. Therefore, the choice of temperature explicitly tips the balance between exploration and exploitation of the loss landscape.

This enables the precise control of learning gradient variance necessary for the proposed study. The variance inherent in $\nabla_\theta \log(p_\theta(\mathbf{x}))$ can be regulated through Thermodynamic Integration, given the term's dependence on $\nabla_\mathbf{z} \log p_\theta(\mathbf{z}|\mathbf{x}, t)$ and $\log p_\theta(\mathbf{z}|\mathbf{x}, t)$.

### 2.3.5 Further motivation for the energy-based prior model

The choice of temperature schedule holds special importance in the context of the latent space energy-based prior model. Beyond its role in constraining gradient noise during training, the temperature schedule directly impacts the dynamic interplay between the GEN and EBM networks.

As detailed in Section 2.2, both the likelihood and prior components are parameterised by neural networks. The EBM-parameterised prior component materialises in Eq. 5 in two distinct ways: firstly, through the latent variables derived from the posterior distribution, which are used to compute the expectation $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\cdot]$; and secondly, through the direct incorporation of the prior term within the likelihood assessment itself, denoted as $p_\alpha(\mathbf{z})$.

However, in the context of Thermodynamic Integration, the role of the prior is significantly reduced, appearing in Eq. 24 solely through the power posterior expectation, denoted as $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x},t)}[\cdot]$, and the Kullback–Leibler divergence terms, denoted by $D_{\mathrm{KL}}(\cdot||\cdot)$.

This implies that gradients associated with the EBM are exclusively tracked through the Langevin sampling loop, used to evaluate samples from $p_\theta(\mathbf{z}|\mathbf{x},t)$, as opposed to the straightforward form depicted for the vanilla model in Eq. 6. This reduces the learning capacity available to the EBM, since it has less opportunity to influence the marginal likelihood evaluation, and completely eliminates the straightforward prior matching formulation detailed in Eq. 6.

Given Eq. 21 and Fig. 5, the use of a temperature schedule characterised by $p > 1$ results in more partitions favouring dependence on the prior. Meanwhile, a temperature schedule characterised by $0 < p \leq 1$ results in partitions stationed closer to the true Bayesian posterior distribution, and therefore more dependence on the likelihood.

Therefore, beyond learning gradient variance, an investigation into the latent space energy-based prior model using Thermodynamic Integration offers a promising avenue to explore the significance of refining latent representations for image modelling. It also presents the opportunity to assess the suitability of training the EBM based off its MCMC sampling influence alone.

## 3 Summary of differences

| Term | Vanilla Model | Altered Model |
|---|---|---|
| $\log(p_\theta(\mathbf{x}))$ | $\mathbb{E}_{p_\theta(\mathbf{z}\|\mathbf{x})}\left[\log(p_\alpha(\mathbf{z})) + \log(p_\beta(\mathbf{x}\|\mathbf{z}))\right]$ | $\int_0^1 \mathbb{E}_{p_\theta(\mathbf{z}\|\mathbf{x},t)}\left[\log p_\beta(\mathbf{x}\|\mathbf{z})\right]\, dt$ |
| $p_\theta(\mathbf{z}\|\mathbf{x},t)$ | $\dfrac{p_\beta(\mathbf{x}\|\mathbf{z})p_\alpha(\mathbf{z})}{\mathcal{Z}(\mathbf{x})}$ | $\dfrac{p_\beta(\mathbf{x}\|\mathbf{z})^t p_\alpha(\mathbf{z})}{\mathcal{Z}(\mathbf{x}\|t)}$ |
| $\nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}\|\mathbf{x},t)$ | $-\nabla_{\mathbf{z}} \dfrac{\|\mathbf{x} - g_\beta(\mathbf{z})\|^2}{2\sigma_l^2} + \nabla_{\mathbf{z}} f_\alpha(\mathbf{z}) - \dfrac{\mathbf{z}}{\sigma_0^2}$ | $-\nabla_{\mathbf{z}} \dfrac{t \cdot \|\mathbf{x} - g_\beta(\mathbf{z})\|^2}{2\sigma_l^2} + \nabla_{\mathbf{z}} f_\alpha(\mathbf{z}) - \dfrac{\mathbf{z}}{\sigma_0^2}$ |
| $\nabla_\alpha \mathcal{L}(\theta, \mathbf{x})$ | Tracked through Langevin sampling for $p_\theta(\mathbf{z}\|\mathbf{x})$ and explicit dependence on $\log(p_\alpha(\mathbf{z}))$, which results in a prior matching formulation in Eq. 6. | Solely tracked through Langevin sampling for $p_\theta(\mathbf{z}\|\mathbf{x},t)$. |

Table 1: Key differences between the vanilla model and the altered model incorporating Thermodynamic Integration.

# 4 Outline of experimental method

## 4.1 Motivation

The following experiments aim to accomplish the following objectives:

1. Cement Thermodynamic Integration as a means of controlling learning gradient variance in a robust, reproducible manner that persists across datasets.
2. Examine the influence of learning gradient variance on the generative potential of the latent space energy-based prior model, and thereby demonstrate why adopting a distribution perspective regarding $\nabla_\theta \mathcal{L}(\theta, \mathbf{x})$ is important.
3. Explore the dynamic interplay between the likelihood parameterised by $\beta$ and the prior parameterised by $\alpha$. Understand how the choice of temperature schedule affects the balance between exploration and exploitation of $\mathcal{L}(\theta, \mathbf{x})$.
4. Explore the discussion outlined in Section 2.3.5, by assessing how significant the energy-based prior correction is in enhancing the overall generative capabilities of the model. Contrast its impact with that of learning gradient noise to gauge their respective influences.
5. Also considering Section 2.3.5, evaluate the suitability of training the EBM solely based on its influence in its short-run MCMC sampling loop for $p_\theta(\mathbf{z}|\mathbf{x}, t)$.

The two experiments conducted to accomplish these are outlined in Sections 4.2 and 4.3. The experimental setup can be found in Section 8.1.

Throughout the experiments, the generative capacity of the models is tracked using image metrics. These have been detailed in Section 4.4.

## 4.2 Overview of experiment 1

To explore the influence of learning gradient variance in the latent space energy-based prior model, we must assess how its training dynamics and generative capabilities vary with respect to the gradient variance.

In our method, we achieve variation in this gradient noise by adjusting either the batch size used or by specifying the value of $p$ in equation 26. Both have been included to cement the potential of Thermodynamic Integration over batching.

All models were trained at fixed complexities, which were reduced from those detailed in [16], until approximate convergence was achieved in both image quality and training gradient variance, (which required 50 epochs). During the training procedures, image samples were generated and characterised using the methods outlined in Section 4.4 to assess generative capacity.

In particular, quantitative metrics capturing the quality of these images were recorded. Steps were taken to ensure that these metrics were as robust and unbiased as possible, as detailed in Section 4.4.

Firstly, the unaltered (vanilla) model was analysed across different batch sizes, $\{25, 50, 75, 150\}$, to assess the effect of batch size on training gradient variance and image fidelity. Subsequently, an altered model incorporating Thermodynamic Integration with varying values of $p$ in Eq. 26 was investigated, maintaining a fixed batch size of 75.

These training loops were repeated five times for each model to ensure experimental rigor, yielding five distinct readings to provide an indication of the uncertainty/robustness of the final gradient

variances and image qualities produced by each model. The experiment was replicated using two datasets, CelebA and CIFAR-10, to confirm the consistency of the observed relationships across different datasets. Both are well-established in current literature and are adopted here to remain consistent. For a detailed description of the experimental setup, please refer to Section 8.1.

### 4.3 Overview of experiment 2

After completing the primary experiment in Section 4.2, a scaled-down secondary experiment was conducted to examine the impact of:

- The amount of temperature discretisation, denoted as $N_t$ in Equation 23
- The number of MCMC iterations required to sample from $p_\theta(\mathbf{z}|\mathbf{x}, t)$, denoted as $K_{\mathbf{z}|\mathbf{x},t}$ in Eq. 12
- The weighting applied to the KL divergence terms of Eq. 24, denoted as $\eta$ in Eq. 27.

This secondary experiment was considered reduced due to its limited execution, with only three repetitions for each training loop, owing to time constraints. Additionally, this secondary experiment focused solely on the $p = 0.1$ model. Despite demonstrating a comparable learning gradient variance to the vanilla models, this model also generated lower fidelity images — a discrepancy we sought to investigate to aid the final two goals outlined in Section 4.1.

We compared against a baseline $p = 0.1$ model using a temperature discretisation of $N_t = 10$, and $K_{\mathbf{z}|\mathbf{x},t} = 20$ MCMC steps to sample from the posterior (Eq. 9). Two alternative models were tested with higher values of these parameters to evaluate the effect on performance. Given time restrictions, only three repetitions were conducted per alteration.

We also introduced a new parameter $\eta$ to weight the $D_{\text{KL}}(\cdot||\cdot)$ bias terms of Eq. 24. This is elaborated further in Section 4.3.3. Increasing this parameter is anticipated to allow the discrepancy between adjacent tempered distributions to exert a larger influence over the loss evaluation. However, the new formulation in Eq. 27 is no longer a true marginal likelihood evaluation.

Specifically a model with $\eta = 2$ is tested for 3 repetitions, to double the emphasis placed on the KL divergence terms in Eq. 24.

#### 4.3.1 Why investigate $K_{\mathbf{z}|\mathbf{x},t}$?

As discussed previously, in the model incorporating Thermodynamic Integration, $\nabla_\alpha \mathcal{L}(\theta, \mathbf{x})$ is solely tracked through the MCMC procedure used to evaluate samples from $p_\theta(\mathbf{z}|\mathbf{x}, t)$.

This tracking is significantly reduced in the altered model with $p = 0.1$, which corresponds to a significant decrease in the influence of the prior, attributed to its specific temperature schedule, (see Fig. 5). This exacerbates the previously discussed disregard for the prior matching formulation in Eq. 6, and the exclusive dependence on Langevin sampling to track the gradients required to update $\alpha$.

This is especially damaging to the EBM's learning capacity. Relying solely on the MCMC sampling procedure for learning gradients presents the following problems:

1. The initial MCMC steps are taken into account, which are inherently noisy and potentially biased. This noise can adversely affect the gradient estimates and subsequently hinder the model's ability to learn accurate exponential tilting. The problem is likely worsened given that short-run MCMC sampling is employed, in accordance with [16].

2. If the MCMC sampler gets stuck in a local mode and fails to explore the other modes effectively, the gradients computed from that sample will be biased towards that particular mode. Consequently, the model may suffer from mode-dropping or mode-collapse issues.

This may be the reason for the reduced generative capacity for the altered model incorporating Thermodynamic Integration, which motivates further study here. If so, it is anticipated that increasing $K_{\mathbf{z}|\mathbf{x},t}$ would result in a less biased sample from $p_\theta(\mathbf{z}|\mathbf{x},t)$, whilst also providing more opportunities for the EBM to learn. However, it is not anticipated to alleviate convergence to a local mode.

#### 4.3.2 Why investigate $N_t$?

As discussed in [3] and Section 2.3.3, increasing the discretisation of the thermodynamic integral decreases the error associated with the KL divergence bias term in Eq. 24. Aside from providing a better estimator for the log-marginal likelihood, this also results in a greater absolute number of partitions of the temperature schedule in regions where the prior is dominant in Eq. 21.

Specifically, the schedule is more finely divided, allowing for more evaluation points to be clustered towards the lower temperatures, as shown in Fig. 5. This provides more opportunities for the prior to influence the learning gradient.

However, more importantly, incorporating more intermediate distributions between the prior and the posterior can aid exploration of the loss landscape, which may help mitigate convergence to local optima. The $p = 0.1$ schedule is significantly more skewed towards the higher temperatures in Fig. 5, which corresponds to $p_\theta(\mathbf{z}|\mathbf{x},t)$ being more representative of the true Bayesian posterior distribution than the exponentially-tilted prior distribution, $p_\alpha(\mathbf{z})$.

The true posterior distribution is expected to be much more complex and multi-modal than $p_\alpha(\mathbf{z})$, which may result in a reduced capacity for exploration using Thermodynamic Integration with a temperature schedule characterised by $0 < p \leq 1$.

These upwards-skewed temperature schedules involve fewer instances of simpler distributions resembling the prior, limiting the model's ability to explore before confronting the more complicated posterior distribution. This may be the reason behind the mode collapse that is clearly evident in Fig. 21.

Increasing $N_t$ may help mitigate the reduced exploration attributed to this temperature schedule, which may enable the $p = 0.1$ model to escape the local optima it has converged to in Fig. 21.

#### 4.3.3 What is $\eta$ and why investigate it?

A new parameter, $\eta$ has been introduced into Eq. 24 to create a new formulation that allows us to examine the impact of the $D_{\text{KL}}(p_{i-1}||p_i)$ in the loss evaluation:

$$\log p_\theta(\mathbf{x}) = \frac{1}{2}\sum_i \Delta t_i(E_{i-1} + E_i) + \eta \cdot \frac{1}{2}\sum_i D_{\text{KL}}(p_{i-1}||p_i) + D_{\text{KL}}(p_i||p_{i-1}) \tag{27}$$

This is no longer a true marginal likelihood evaluation, but it allows us to control the weighting applied to the $D_{\text{KL}}(p_i||p_{i-1})$ terms, which corresponds to the error associated with the trapezium rule for numerical integration.

Here, the KL divergences represent the discrepancies between adjacent tempered distributions in the temperature schedule (Eq. 21). By increasing the emphasis on these discrepancies relative to the expected likelihood terms in Eq. 24, the influence of the posterior distribution terms in the loss evaluation, (in which the prior is manifested), may be enhanced, thereby improving the EBM's learning capacity.

More significantly, we can apply more weight to the discrepancies between adjacent tempered distributions, allowing us to investigate the balance between exploration and exploitation that the model is faced with. These discrepancies may correspond to the appearance of local modes and added complexity within the intermediary distributions between the prior and the true Bayesian posterior distributions.

Therefore, if increasing $\eta$ corresponds to better image quality, we may be able to attribute the poor performance of the $p = 0.1$ model to the imbalanced exploration and exploitation. The model may have been unable to adjust itself between adjacent temperature schedules because the initial change in complexity between adjacent distributions is too large, as indicated by the initial rise in Fig. 5 for $p = 0.1$. Here, the Thermodynamic Integration loop quickly skips past the earlier, simpler intermediary distributions and immediately begins to explore the more complicated posterior distribution.

A higher value of $\eta$ may force the model to better navigate the discrepancies and increasing complexity as it transitions from the prior to the true posterior, potentially improving performance if the default balance between exploration and exploitation is suboptimal.

While this inclusion alters the true marginal likelihood into a form that only resembles it, the model should still be able to learn the data-space by minimising Eq. 27. However, it is also anticipated that the variance of the learning gradients may increase as a result of this scalar multiplication.

### 4.4 Tracking image fidelity

Quantifying the perceptual quality of an image dataset poses a significant challenge, leading to the emergence of multiple metrics across machine learning literature. However, none of these metrics are flawless. They each come with their own equations, such as those depicted in 28 and 29, and introduce biases that are contingent on the number of samples used. For instance, computing the mean in Eq. 28 necessitates a Monte Carlo estimate, which carries an associated error as shown in Eq. 18.

Among the metrics widely embraced by the community are the Fréchet Inception Distance (FID) [7] and the Kernel Inception Distance (KID) [1]. These metrics, detailed underneath, are commonly employed despite their limitations.

To extract feature representations from images, the InceptionV3 model was used, a convolutional neural network (CNN) pre-trained on the ImageNet dataset [19]. In particular, for the JAX implementation of this study, the pre-trained InceptionV3 implementation was taken from the work of [13].

When an image is inputted into InceptionV3, it undergoes multiple layers of convolutional and pooling operations, thereby capturing features at various levels of abstraction. These extracted features are subsequently used in the equations provided below to compute the metrics. Lower metric values correspond to higher quality generations.

#### 4.4.1 FID

$$\text{FID} = \|\mu_{\text{real}} - \mu_{\text{gen}}\|^2 + \text{Tr}(\Sigma_{\text{real}} + \Sigma_{\text{gen}} - 2\sqrt{\Sigma_{\text{real}}\Sigma_{\text{gen}}}) \tag{28}$$

Here, $\mu_{\text{real}}$ and $\mu_{\text{gen}}$ represent the mean feature representations of real and generated samples, respectively, obtained via InceptionV3. Similarly, $\Sigma_{\text{real}}$ and $\Sigma_{\text{gen}}$ denote their respective covariance matrices.

#### 4.4.2 KID

$$\text{KID} = \frac{1}{n^2}\text{Tr}(H\tilde{H}) - \frac{2}{n}\text{Tr}(HK) + \text{Tr}(KK) \tag{29}$$

where $n$ is the number of samples, $H$ is the Gram matrix computed from the feature representations of real samples, $\tilde{H}$ is the Gram matrix computed from the feature representations of generated samples, and $K$ is the cross-gram matrix between real and generated samples.

Here, the Gram matrix $\tilde{H}$ was computed using the radial basis function (RBF) kernel. Each entry $\tilde{H}_{ij}$ of the matrix is computed as the RBF kernel function applied to the Euclidean distance between the feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Specifically, for $\tilde{H}$, the RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ with length scale $\gamma = \frac{1}{\text{num\_features}}$ was used:

$$\tilde{H}_{ij} = \exp\left(-\frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2}{2\gamma^2}\right)$$

Here, $\|\cdot\|^2$ represents the squared Euclidean distance between the feature vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ of the generated images, obtained from the InceptionV3 model.

#### 4.4.3 Unbiased image metrics for a reliable experimental method

To gauge the perceptual quality of images generated by the latent space energy-based model in a manner resilient to the limitations outlined earlier, both the FID and the KID metrics are tracked. This dual approach allows for a more comprehensive assessment of image quality while acknowledging and mitigating the biases inherent in each metric.

However, it was ultimately discovered that KID was the more reliable metric, as detailed later in this Section. Therefore, FID was disregarded in the interpretation of results within Section 5.

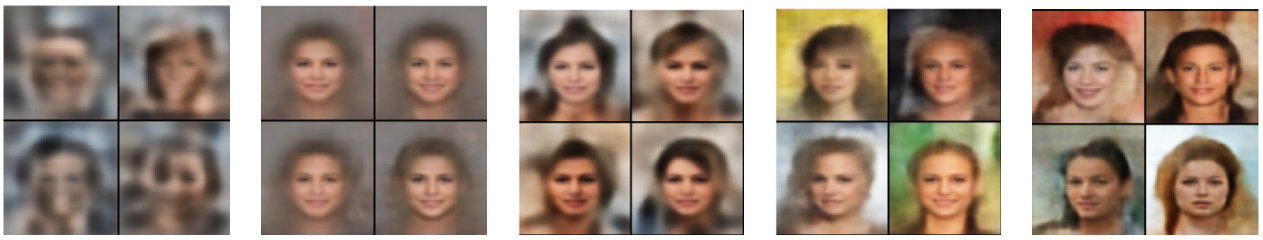
Furthermore, to address the bias in these metrics as mentioned earlier, a scheme inspired by the prior work of [4] is adopted. In this approach, FID readings are gathered for various sample sizes, and then linear regression is used to extrapolate to an infinitely-sized sample set. This technique yields effectively unbiased estimators of the metrics, enhancing the robustness of the perceptual quality evaluation process.

In our experiment, to implement this scheme for both FID and KID, the following sample set sizes are employed:

$$\text{Samples Sizes} : \{1000, 1389, 1778, 2167, 2556, 2944, 3333, 3722, 4111, 4500\}$$

Linear regression is subsequently applied to evaluate the unbiased estimators, $\overline{FID}_\infty$ and $\overline{KID}_\infty$. In general, $\overline{KID}_\infty$ emerged as the more robust metric across all datasets. Figs. 6 and 7 present visual examples to illustrate this observation.

Specifically, throughout the experiments detailed in Sections 4.2 and 4.3, generative performance and their relation to the metrics was verified through human inspection of recorded image samples. In Figs. 6 and 7, each grid of images corresponds to different models at various stages of their training runs, each chosen to exemplify how the metrics assess the perceptual quality of different images. For both datasets, the models used to generate samples were of substantially reduced complexity compared to the original implementation in [16].



(a) $\overline{FID}_\infty = 380.5$, $\overline{KID}_\infty = 0.102$ (b) $\overline{FID}_\infty = 246.3$, $\overline{KID}_\infty = 0.075$ (c) $\overline{FID}_\infty = 268.1$, $\overline{KID}_\infty = 0.067$ (d) $\overline{FID}_\infty = 275.0$, $\overline{KID}_\infty = 0.050$ (e) $\overline{FID}_\infty = 252.5$, $\overline{KID}_\infty = 0.033$

Figure 6: Performance of unbiased metrics against samples of the CelebA dataset, generated by various implementations of the vanilla and altered models at different stages of their 50-epoch training runs. This collection of chosen images serves to illustrate why $\overline{KID}_\infty$ is considered to be more reliable. For instance, in Fig. 6b, despite exhibiting the lowest $\overline{FID}_\infty$, suggesting superior image quality, the grid of images is subjectively less impressive than Figs. 6c, 6d, and 6e. In contrast, $\overline{KID}_\infty$ was more representative of what can be considered subjectively better quality.

(a) $\overline{FID}_\infty = 492.7$, $\overline{KID}_\infty = 0.084$ (b) $\overline{FID}_\infty = 456.2$, $\overline{KID}_\infty = 0.074$ (c) $\overline{FID}_\infty = 464$, $\overline{KID}_\infty = 0.052$ (d) $\overline{FID}_\infty = 454.2$, $\overline{KID}_\infty = 0.046$ (e) $\overline{FID}_\infty = 459$, $\overline{KID}_\infty = 0.039$

Figure 7: Performance of unbiased metrics against samples of the CIFAR-10 dataset, generated by our different implementations of the vanilla and altered models at different stages of their 50-epoch training runs.. Generally, with reduced complexity, the model struggled with the diversity of CIFAR-10. However, once again, the provided examples subjectively demonstrate why $\overline{KID}_\infty$ is considered more reliable for this study. For example, Fig. 7b is attributed a fairly low $\overline{FID}_\infty$, despite being comparatively worse in quality than Figs. 7c, 7d, and 7e.

## 5 Results

### 5.1 Experiment 1 results

The results below are shown for the experiment outlined in Section 4.2. This experiment seeks to cement Thermodynamic Integration as a robust means of controlling variance, as well as investigate the effect of learning gradient variance on image fidelity.

#### 5.1.1 Controlling the gradient variance

Figures 8 and 9 showcase the different learning gradient variances achievable by the trained vanilla and altered models respectively. The general relationship persisted for CIFAR-10, provided in Section 8.3.2.

The boxplots demonstrate that different variances can be achieved by adjusting batch sizes, $N_{batch}$, for the vanilla model or by varying temperature powers, $p$, for the altered model.

However, the ranges of the variances observed between the five distinct repetitions is notably wider in Fig. 8 compared to Fig. 9. The altered model incorporating Thermodynamic Integration can therefore achieve learning gradient variances comparable to those achieved by adjusting batch sizes in the vanilla model, but with greater reliability across repetitions.

This means that Thermodynamic Integration offers a more robust and controllable means of achieving different learning gradient variances through careful control of the temperature schedule.

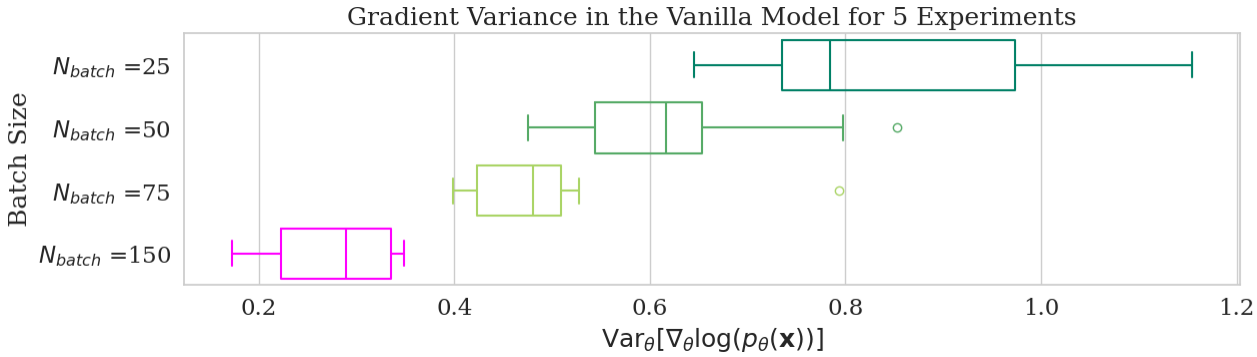



Figure 8: Relationship between final learning gradient variance of the vanilla model and batch size. The models were trained on CelebA for 50 epochs.

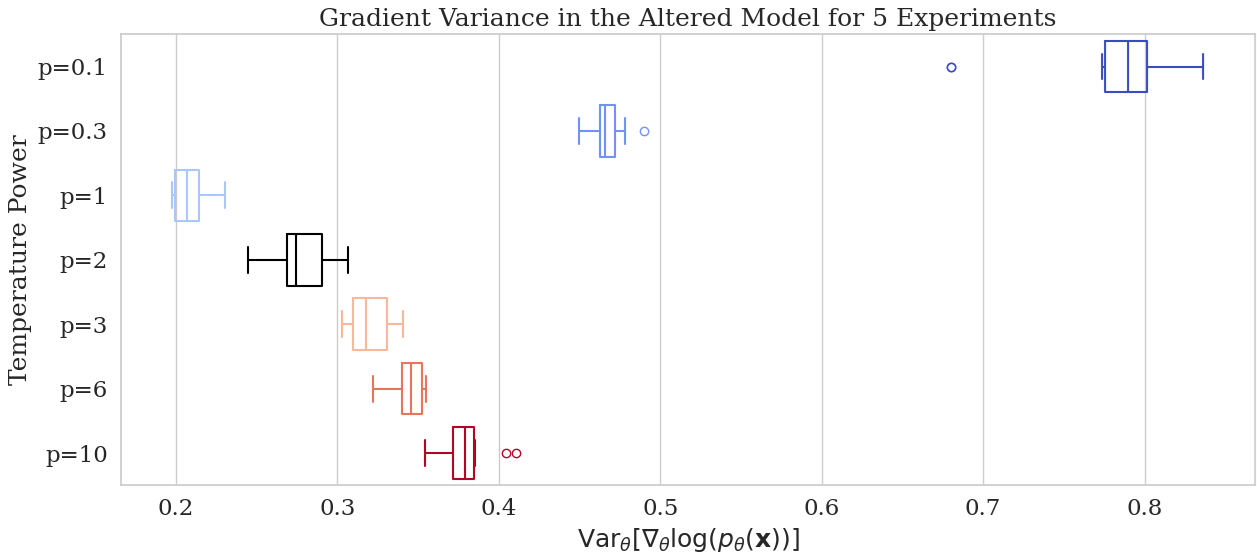



Figure 9: Relationship between final learning gradient variance and $p$ in Eq. 26 for the altered model. The models were once again trained on CelebA for 50 epochs. The amount of discretisation (Eq. 23) is held fixed at $N_t = 10$. Larger variances are attainable with $0 < p \leq 1$. Fine-grained changes can be achieved with $p > 1$.

### 5.1.2 Gradient variance and image fidelity

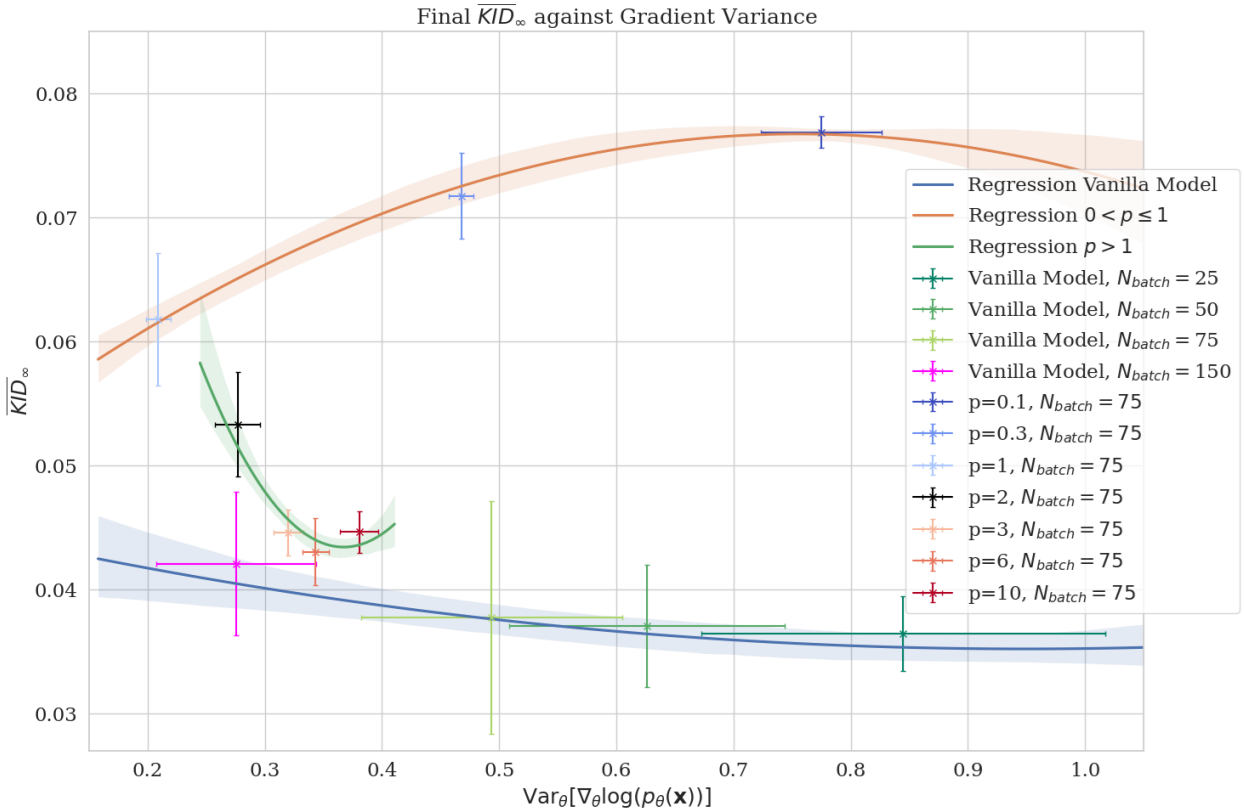



Figure 10: The relationships between learning gradient noise and generated image fidelity, as described by $\overline{KID}_\infty$. These results correspond to models trained on the CelebA dataset for 50 epochs. Quadratic regressions have been included to serve as fitting representations of the different regimes. Lower $\overline{KID}_\infty$ corresponds to higher quality generated samples. The means across repetitions are plotted, with error bars representing the standard deviations.

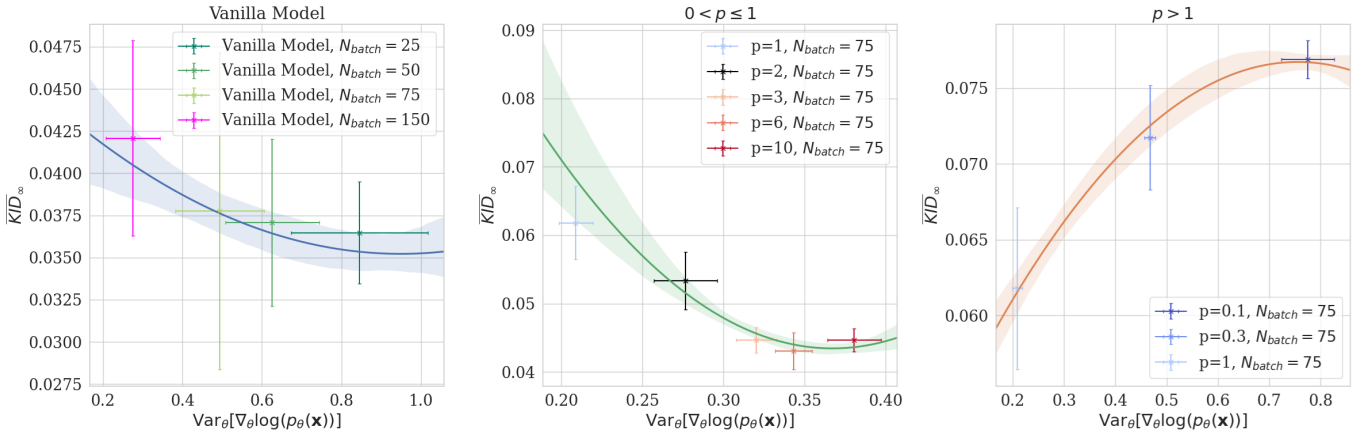



Figure 11: The relationships between learning gradient noise and generated image fidelity, as quantified by $\overline{KID}_\infty$ for CelebA. This plot, equivalent to Fig. 10, has been separated into its different regimes for more clarity.

Shown in Figs. 10 and 11 are the results for the different vanilla and altered models incorporating the experimental setups described in Section 8.1. In the vanilla model, batch size was used to tune learning gradient variance. In the altered model incorporating Thermodynamic Integration, the batch size was held at 75, whilst $p$ in Eq. 26 was used to alter learning gradient variance.

In terms of image fidelity, the vanilla model outperformed the altered model, despite the large computational cost incurred by the adoption of Thermodynamic Integration, as explained in Section 8.3.1. Section 5.2 illustrates that increasing $N_t$ can improve image quality, but comes with an even greater computational cost, also outlined in Section 8.3.1. This suggests that Thermodynamic Integration should be used solely when precise control over learning gradient variance is necessary rather than image quality.

In the $p > 1$ regime, the altered model attains comparable learning gradient variances as the vanilla model with $N_{\text{batch}} = 150$. Further increasing $p$ beyond 1 increases the learning gradient variance, and seemingly enhances generative capacity, (resulting in lower $\overline{KID}_\infty$), until a minimum is achieved at $p = 6$.

Quadratic regressions have been applied to the results. Overall, they show that there is a striking dependence between image fidelity and gradient variance. However, the relationship must first be divided into different regimes depending on model architecture.

They also show that generative capacity is generally improved with larger learning gradient variances for the vanilla models. Although cross-referencing with Figs. 18 and 19 in the Appendix suggests that this trend holds only up to a certain variance, beyond which increasing gradient noise deteriorates image quality. Similarly for the altered models with $p > 1$, a particular variance is required to achieve a clear maximum image fidelity, (around 0.35).

Importantly, a large discrepancy arises between the models in Fig. 10 when the altered model operates within the $0 < p \leq 1$ regime. For comparable variances in Fig. 10, the altered model yields lower-quality images than the vanilla model . This suggests that learning gradient variance is not the sole influence regarding the model's generative capacity.

The boxplots of Fig. 9 suggests that to achieve large variances, temperature powers within the range of $0 < p \leq 1$ are required. This reduces the role of the EBM-parameterised prior within marginal likelihood evaluations, as elaborated in Section 2.3.5. By diminishing the influence of the prior, the learning capacity of the EBM model is correspondingly diminished, as the tracking of $\nabla_\alpha \mathcal{L}(\theta, \mathbf{x})$ is confined solely to the Langevin sampling loop for $p_\theta(\mathbf{z}|\mathbf{x}, t)$, which is required to evaluate $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x},t)}[\cdot]$ and $D_{\text{KL}}(\cdot||\cdot)$ in Eq. 24.

It may also result in a reduced capacity for exploration, as the thermodynamic integral is characterised by having more evaluations skewed towards higher temperatures, also elaborated in Section 2.3.5. This favours the true posterior distribution over the prior, pushing the loss evaluation towards exploitation rather than exploration, given the more complex form of the true Bayesian posterior distribution.

This may be the reason for the discrepancy between the vanilla and altered models incorporating $p$ in the range, $0 < p \leq 1$. A standout example is $p = 0.1$, which exhibits a similar learning gradient variance as the vanilla model with $N_{batch} = 25$, but demonstrates comparably worse images and a larger $\overline{KID}_\infty$, which stands as motivation for the experiment outlined in Section 4.3.

Cross-referencing against Fig. 21 in the Appendix suggests that this discrepancy is attributed to mode collapse. The altered models with $0 < p \leq 1$ converged to a local optima rather than the true minimum in the loss landscape. This is explored in more detail through the experiment outlined in Section 4.3.

## 5.2 Experiment 2 results

Section 4.3 outlines the details of the secondary experiment, for which the results are presented below. This experiment seeks to explain the discrepancy observed in the results of the previous experiment.

The discrepancy corresponds to a mode collapse for the models operating with $0 < p \leq 1$, as shown in Fig. 21. To do this, $N_t$, $K_{\mathbf{z}|\mathbf{x},t}$, and $\eta$ are increased for the reasons discussed in Section 4.3.

### 5.2.1 Explaining the discrepancy

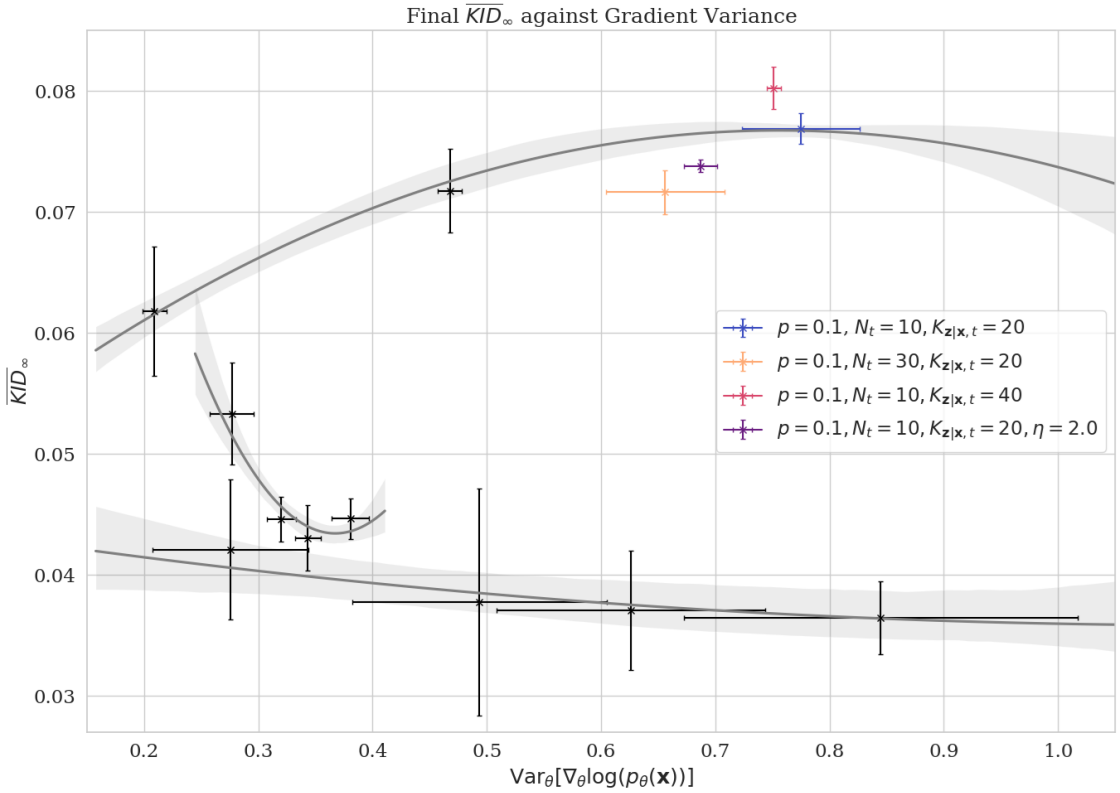



Figure 12: The models with the alterations discussed in Section 4.3 are overlaid on the previous data for comparison. The model with $p = 0.1$, $N_t = 10$, $K_{\mathbf{z}|\mathbf{x},t} = 20$ is carried over from Fig. 10 to serve as a baseline. The means across repetitions are plotted, with error bars representing the standard deviations. The previous data comprises 5 repetitions, whilst the new data only comprises 3.

Presented in Fig. 12 are the new results overlaid on the previous results for reference. The final images that these models were able to generate are provided in Fig. 13, which also includes a $p = 6$ model for reference.

Notably, all $p = 0.1$ models still suffered from mode collapse, which suggests that the discrepancy is a result of the form of the $p = 0.1$ temperature schedule in Fig. 5 rather than any tunable parameters. Analysing the observations from Fig. 12 can concretely explain this.

Firstly, increasing $K_{\mathbf{z}|\mathbf{x},t}$ from 20 to 40 resulted in similar, if not worse, image quality in Fig. 12. Following the discussion in Section 4.3.1, this suggests that the EBM's learning capacity is not limited by the MCMC sampling procedure.

The mode collapse evident in Fig. 21 and Fig. 12 for the $p = 0.1$ model is therefore unlikely to be attributed to poor exploration or bias of the unadjusted Langevin algorithm. Moreover, it implies that the poor performance of the $p = 0.1$ model cannot be solely attributed to gradient tracking being confined to the MCMC sampling steps, as this effect should have been somewhat mitigated by increasing $K_{\mathbf{z}|\mathbf{x},t}$, which would reduce the bias manifesting in samples from $p_\theta(\mathbf{z}|\mathbf{x}, t)$, whilst providing more opportunities for the EBM to learn.

Increasing the schedule discretisation parameter, $N_t$, and the weighting applied to the $D_{\text{KL}}(\cdot||\cdot)$ terms, $\eta$, improved image fidelity and reduced the variance of learning gradients. The improvement is more significant with increased $N_t$ than with increased $\eta$, though both factors seem to contribute positively. The improvement in image fidelity, despite the reduction in learning gradient variance, reinforces the claim that learning gradient variance is not the sole contributor to image fidelity.

In line with the discussion presented in Sections 4.3.2 and 4.3.3, this suggests that the generative capacity of the altered model is directly dependent on the balance between exploration and exploitation provided by the temperature schedule.

A larger $N_t$ allows the model to step through more intermediary distributions between the simpler prior and more complex posterior distributions, enhancing its ability to explore the loss landscape.

Increasing the weighting of the discrepancy terms in Eq. 27 with $\eta$ may have emphasised the added complexity or local modes between adjacent tempered distributions, potentially causing the model to navigate them more aggressively. However, the limited impact of introducing $\eta$, (as shown in Fig. 12), might simply be due to random error as a result of the limited experiment count.

Increasing $N_t$ emerges as the most promising approach towards enhancing image fidelity in the model incorporating Thermodynamic Integration, albeit at a considerable cost as depicted in Fig. 14. On the other hand, increasing $\eta$ yields a less pronounced positive effect without incurring additional computational overhead but also leads to an inexact marginal likelihood evaluation.

However, in both cases, the model was unable to escape mode collapse, as evident in the generated samples shown in Fig. 13. This may be due to two reasons:

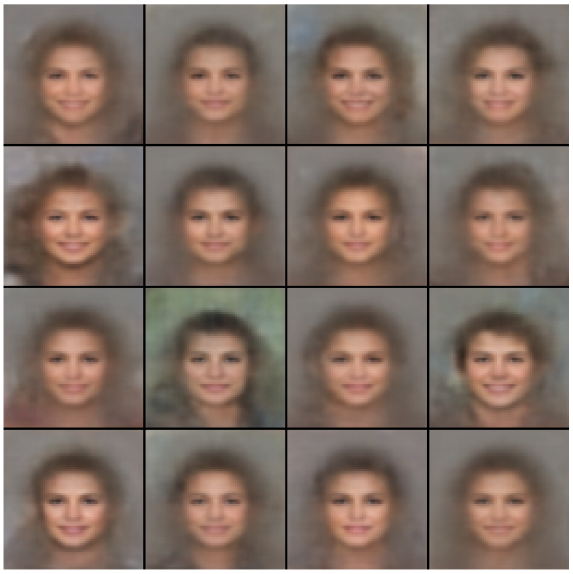
- The parameters $N_t$ and $\eta$ were only increased slightly. Perhaps a much larger discretisation of the temperature schedule is needed to allow the model to escape the local minima entirely.

- The parameters $N_t$ and $\eta$ may only be suitable for fine-tuning image fidelity once a minimum in the loss landscape has been converged to. The underlying issue of mode collapse might only be solvable by a temperature schedule that favors partitions clustered towards the simpler prior distribution, i.e., $p > 1$.

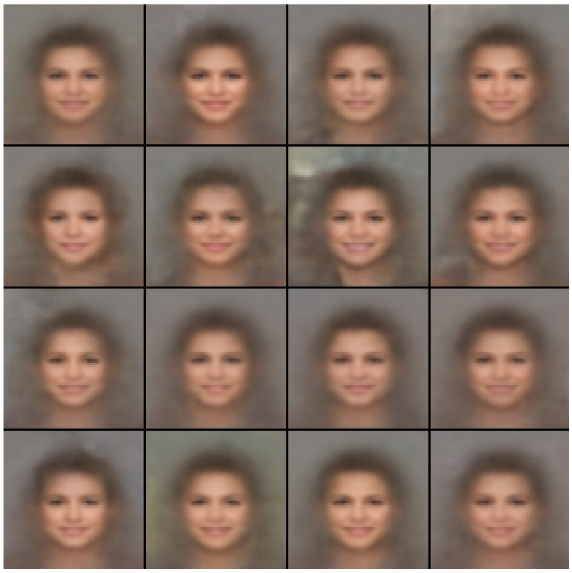
Overall, the results suggest that compared to the temperature schedule governed by $p$, varying $K_{\mathbf{z}|\mathbf{x},t}$ and $\eta$ produced statistically insignificant results. More repetitions and larger degrees of variation are required to determine if they have any significant impact on image fidelity.

In contrast, the temperature discretisation $N_t$ had a statistically significant impact on improving and fine-tuning image quality, but did not help the model escape local optima. Larger values for $N_t$ are likely to further enhance the model's explorative capacity and perhaps even help it escape local modes. However, this comes at a high cost, as shown in Fig. 14, where the training time for 50 epochs increased by a factor of 3 from the baseline $p = 0.1$ model with the increase from $N_t = 10$ to $N_t = 30$.

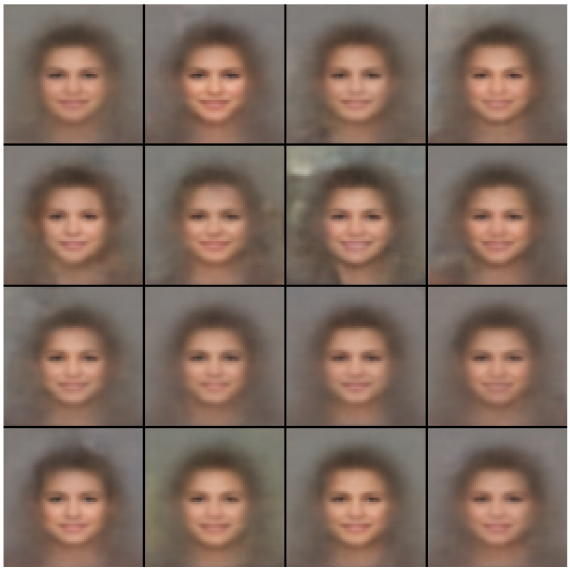
(a)
$p = 6, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 20$



(b)
$p = 0.1, N_t = 30, K_{\mathbf{z}|\mathbf{x},t} = 20$



(c)
$p = 0.1, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 40$



(d)
$p = 0.1, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 20, \eta = 2.0$

Figure 13: CelebA images generated by the models of Fig. 12. All $p = 0.1$ models have suffered from mode collapse, in comparison to the $p = 6$ model, which has been included as reference.

### 5.3 Summary of findings

#### 5.3.1 Learning gradient variance and image fidelity

| Param. | Learning Gradient Noise | Image Fidelity | Controllable? |
|---|---|---|---|
| $N_{batch}$ | Decreasing $N_{batch}$ increases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | Decreasing $N_{batch}$ improves quality until a minimum is achieved, (depending on the dataset being learned). | No - large range across repetitions |
| $p > 1$ | Increasing $p$ increases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | Increasing $p$ improves quality until a minimum is achieved at $p \approx 6$. Increasing further reduces quality. | Yes - small range across repetitions |
| $0 < p \leq 1$ | Decreasing $p$ increases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | Much poorer image quality due to mode collapse. Increasing $p$ improves quality | Yes - small range across repetitions |

Table 2: A summary regarding how learning gradient noise and image fidelity are affected by the batch size and temperature schedule.

#### 5.3.2 Importance of Thermodynamic Integration parameters

| Param. | Learning Gradient Noise | Image Fidelity | Significance | More Work? |
|---|---|---|---|---|
| $p$ | (See Tab. 2) | (See Tab. 2) | Upmost significance - defines the critical balance between exploration and exploitation | Different forms of temperature scheduling could be explored |
| $N_t$ | Increasing $N_t$ decreases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | Increasing $N_t$ improves quality | Fair significance - improves model performance, but at a large cost (Fig. 14). | $N_t >> 30$ should be investigated to determine if $N_t$ is only suitable for fine-tuning or is more significant entirely |
| $K_{\mathbf{z}\|\mathbf{x},t}$ | Increasing $K_{\mathbf{z}\|\mathbf{x},t}$ decreases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | No impact or increasing $K_{\mathbf{z}\|\mathbf{x},t}$ may even worsen image quality | More experimental work is required, but the results suggest that short run MCMC is suitable enough. Increasing $K_{\mathbf{z}\|\mathbf{x},t}$ comes at a computational cost (Fig. 14) | A larger number of repetitions is required to improve reliability. Metropolis-adjusted Langevin sampling could be investigated, provided that differentiability is maintained. |
| $\eta$ | Increasing $\eta$ decreases $\mathrm{Var}_\theta\left[\nabla_\theta \mathcal{L}(\theta, \mathbf{x})\right]$ | Increasing $\eta$ slightly improves quality at no computational cost | More experimental work is required, but the results suggest that $\eta$ may be a promising means of forcing the model to navigate adjacent tempered distributions more carefully | A larger number of repetitions is required to improve reliability. |

Table 3: A summary regarding how the different parameters of the altered model affect learning gradient noise and image fidelity.

# 6 Reflection

## 6.1 Limitations of the experimental method

While efforts were made to maintain the rigor of the experimental method, including repetition, testing across two datasets, and the selection and validation of unbiased image metrics, several weaknesses remain that could have potentially compromised the study's completeness or reliability.

### 6.1.1 Number of repetitions

The limited number of repetitions conducted, (5 for the first experiment and 3 for the second), may have impacted the reliability and robustness of the results. When dealing with small sample sizes, variance of the mean can be relatively high, leading to greater uncertainty regarding the precise locations of the means in Figs. 10 and 12.

Additionally, the small number of repetitions may have failed to fully capture the true population variance, and the influence of any outliers from the true distributions may have been amplified, potentially leading to biased conclusions.

To enhance the reliability of the results, more repetitions should have been used, e.g. 10 repetitions per training loop. This would have helped to reduce the variance of the mean, provide a more accurate representation of the population distribution, and minimise the impact of random chance on the observed outcomes.

### 6.1.2 Reduced model complexity

The model complexity used for all datasets and experiments was reduced from the one presented in [16]. Furthermore, the architectures used in this study differ from those used in the previous work, as presented in Section 8.1.

Specifically, the U-Net decoder architecture used for the generator in [16] has been replaced with a constant kernel size convolutional neural network architecture here. These modifications were implemented to accelerate the experimental process, enabling the collection of a sufficient number of results within a reasonable timeframe.

However, by opting for a simpler architecture, it is likely that some potential generative capacity has been sacrificed. Given that the findings are based on comparative observations, this is not anticipated to be a limiting factor. Nonetheless, conducting the same experiment with the setup presented in [16] may reveal contrasting findings.

### 6.1.3 Limited datasets

The experiments were conducted on two datasets, CelebA and CIFAR-10. While this provides some diversity, evaluating the models on additional datasets, particularly those with different complexities or characteristics such as SVHN, could further validate the findings.

### 6.1.4 Hyperparameter tuning

The hyperparameter choices detailed in Section 8.1 were adopted from [16]. However, given the architectural changes and the introduction of the model incorporating Thermodynamic Integration, these choices may have been suboptimal for the current study. Despite this, the findings remain reliable as they are based on comparative observations and were consistent across different datasets.

Nevertheless, hyperparameter tuning could potentially affect the results, for instance, by alleviating the mode collapse observed in Fig. 21 for the models operating in the $0 < p \leq 1$ regime. A more thorough exploration of hyperparameter settings to optimise performance and address such issues may have improved the reliability of the image metrics.

#### 6.1.5 Limited time

Some aspects of the experiments were limited due to time constraints, such as the number of repetitions, hyperparameter tuning, and the values of certain parameters (e.g., $N_t$). Access to more computational resources could have potentially alleviated some of these constraints and enabled more extensive experimentation.

### 6.2 Future work

In addition to addressing the limitations outlined in Section 6.1 and expanding the experiments by gathering more data with additional repetitions to further validate the findings presented in Section 5, future work could additionally explore the vast potential of Thermodynamic Integration.

#### 6.2.1 Temperature scheduling

Exploring different forms of temperature scheduling beyond the power law formulation of Eq. 26 used in this study could help concretely identify the cause of mode collapse observed in the models operating within the $0 < p \leq 1$ regime. Alternative scheduling strategies may provide better control over the exploration-exploitation trade-off and potentially enhance the model's performance.

#### 6.2.2 Metropolis-adjusted MCMC

Investigating the use of Metropolis-adjusted Langevin sampling instead of the unadjusted Langevin algorithm employed in this work may also help mitigate the mode collapse issue observed for certain temperature schedules, although issues regarding differentiability may arise.

#### 6.2.3 Other generative models

Investigating the application of Thermodynamic Integration and learning gradient variance control to other generative models, such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models could help verify the consistency of the findings and further demonstrate the potential of Thermodynamic Integration in controlling gradient noise. Sequence modelling could also be explored in addition to image modelling.

#### 6.2.4 Improving Thermodynamic Integration

Exploring the integration of Thermodynamic Integration with other techniques aimed at enhancing generative model performance, such as adversarial training, regularisation methods, or curriculum learning strategies may yield promising insights. There may be potential for improving the exploration of loss landscapes and fine-tuning gradient noise control through this method.

Additionally, investigating how Thermodynamic Integration and gradient variance control enhances various aspects of generative models beyond traditional performance metrics might be an avenue worth exploring. Examples include improvements to sample diversity, mode coverage, and controllability. Learning gradient noise could prove well suited towards predicting them in a robust manner.

## 7 Conclusions

1. The first experiment successfully demonstrated that Thermodynamic Integration can control learning gradient variance in a reproducible manner. It demonstrated greater consistency across repetitions compared to batching and provided more precise, arbitrary control over the variance than simply minimising it using alternate methods such as Gradient Clustering [5].
2. The experiment also revealed that, in the absence of mode collapse, increasing the learning gradient noise improved image quality up to an optimal point. Beyond this maximum image fidelity, further increasing the variance worsened the image quality.
3. Despite this clearly defined and consistent relationship between image fidelity and learning gradient variance, the results varied depending on the type of model used and the temperature schedule adopted. Overall, the experiment revealed that learning gradient variance alone is insufficient to predict model performance in image generation.
4. The second experiment explored the dynamic interplay and complexities between the true Bayesian posterior distribution and the exponentially-tilted prior distribution. It highlighted the critical importance of the temperature schedule in Thermodynamic Integration, showcasing its impact on exploring and exploiting the loss landscape. The experiment also demonstrated how tuning the parameters governing Thermodynamic Integration can mitigate the adverse effects of an imbalance between exploration and exploitation.
5. The outcomes of the first experiment suggested that energy-based prior correction is crucial for the model's performance. Clustering more temperatures in favor of this prior distribution generally enhanced image fidelity.
6. The second experiment demonstrated that short-run MCMC sampling is likely sufficient for training the EBM, as performance did not improve with a larger number of MCMC steps.
7. Given that the training loops were repeated multiple times, these findings are fairly statistically rigorous. Moreover, since the results persisted across both the CelebA and CIFAR-10 datasets, it is likely that the outcomes are universally applicable. However, the limitations of the experimental methods adopted were discussed, indicating that further investigations are needed to validate the findings with greater statistical rigor.

Overall, this study demonstrates the importance of adopting a distributional perspective on the learning gradient to enhance the capabilities of deep generative models and the quality of the images they produce. It presents Thermodynamic Integration as a powerful method for controlling learning gradient variance and balancing exploration and exploitation by altering the intermediary tempered distributions explored by MCMC methods.

However, it also shows that while Thermodynamic Integration provides these advantages, it comes with a significant computational cost and does not achieve better image fidelity compared to traditional methods of estimating the marginal likelihood, especially when limited schedule discretisation is used.

## References


[1] Mikołaj Bińkowski et al. *Demystifying MMD GANs*. 2021. arXiv: 1801.01401 [stat.ML].

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. 2019. arXiv: 1809.11096 [cs.LG].

[3] Ben Calderhead and Mark Girolami. "Estimating Bayes factors via thermodynamic integration and population MCMC". In: *Computational Statistics & Data Analysis* 53.12 (2009), pp. 4028–4045. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2009.07.025. URL: https://www.sciencedirect.com/science/article/pii/S0167947309002722.

[4] Min Jin Chong and David Forsyth. *Effectively Unbiased FID and Inception Score and where to find them*. 2020. arXiv: 1911.07023 [cs.CV].

[5] Fartash Faghri et al. *A Study of Gradient Variance in Deep Learning*. 2020. arXiv: 2007.04532 [cs.LG].

[6] N. Friel and A. N. Pettitt. "Marginal Likelihood Estimation via Power Posteriors". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 70.3 (2008), pp. 589–607. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/20203843 (visited on 10/26/2023).

[7] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *CoRR* abs/2006.11239 (2020). arXiv: 2006.11239. URL: https://arxiv.org/abs/2006.11239.

[9] Seung-Seop Jin, Heekun Ju, and Hyung-Jo Jung. "Adaptive Markov chain Monte Carlo algorithms for Bayesian inference: recent advances and comparative study". In: *Structure and Infrastructure Engineering* (June 2019). DOI: 10.1080/15732479.2019.1628077.

[10] Hesaneh Kazemi, Carolyn C. Seepersad, and H. Alicia Kim. "Multiphysics Design Optimization via Generative Adversarial Networks". In: *Journal of Mechanical Design* 144.12 (Oct. 2022), p. 121702. ISSN: 1050-0472. DOI: 10.1115/1.4055377. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/144/12/121702/6934726/md_144_12_121702.pdf. URL: https://doi.org/10.1115/1.4055377.

[11] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[12] Elizabeth Koehler, Elizabeth Brown, and Sebastien Haneuse. "On the Assessment of Monte Carlo Error in Simulation-Based Statistical Analyses". In: *The American statistician* 63 (May 2009), pp. 155–162. DOI: 10.1198/tast.2009.0030.

[13] Matthias Wright and Hayden Donnelly and Boris Dayma and Saurav Maheshkar. *jax-fid: FID computation in Jax/Flax*. https://github.com/matthias-wright/jax-fid. License: Apache-2.0 License. Year of Last Update.

[14] François Mazé and Faez Ahmed. *Diffusion Models Beat GANs on Topology Optimization*. 2022. arXiv: 2208.09591 [cs.LG].

[15] Arvind Neelakantan et al. *Adding Gradient Noise Improves Learning for Very Deep Networks*. 2015. arXiv: 1511.06807 [stat.ML].

[16] Bo Pang et al. "Learning Latent Space Energy-Based Prior Model". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21994–22008. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/fa3060edb66e6ff4507886f9912e1ab9-Paper.pdf.

[17] Corey Parrott, Diab Abueidda, and Kai James. "Multi-Head Self-Attention GANs for Multiphysics Topology Optimization". In: June 2022. DOI: 10.2514/6.2022-3726.

[18] Herbert E. Robbins. "A Stochastic Approximation Method". In: *Annals of Mathematical Statistics* 22 (1951), pp. 400–407. URL: https://api.semanticscholar.org/CorpusID:16945044.

[19] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[20] Aysim Toker et al. *SatSynth: Augmenting Image-Mask Pairs through Diffusion Models for Aerial Semantic Segmentation*. 2024. arXiv: 2403.16605 [cs.CV].

# 8 Appendix

## 8.1 Experiment setup

As mentioned previously, the experiments are hosted at `https://github.com/PritRaj1/JAX-ThermoEBM`. The raw experimental readings have also been provided as part of the codebase.

### 8.1.1 Hyperparameters

| Param. | Codebase Var. Name | Value |
|---|---|---|
| Dataset | DATASET | CelebA |
| Number of training epochs | NUM_EPOCHS | 50 |
| Batch size | BATCH_SIZE | 75 (variable for vanilla model) |
| Number of training examples | NUM_TRAIN_DATA | 12000 |
| Number of testing examples | NUM_VAL_DATA | 4500 |
| Number of repetitions to conduct | NUM_EXPERIMENTS | 5 |
| Latent space dimension | Z_CHANNELS | 80 |
| Complexity of EBM, ($nez$ in tab. 6) | EBM_FEATURE_DIM | 100 |
| EBM Leaky-ReLU leak coefficient | EBM_LEAK | 0.1 |
| Complexity of GEN, ($ngz$ in tab. 6) | GEN_FEATURE_DIM | 64 |
| GEN Leaky-ReLU leak coefficient | GEN_LEAK | 0.2 |
| $\sigma_0$ (for $\pi_0(\mathbf{z})$ in Eq. 1) | p0_SIGMA | 1 |
| $\sigma_l$ (for $\epsilon$ in eq 2) | LKHOOD_SIGMA | 0.3 |
| Initial learning rate for $\alpha$ | E_INITIAL_LR | 0.00002 |
| Initial learning rate for $\beta$ | G_INITIAL_LR | 0.0001 |
| Final learning rate for $\alpha$ | E_FINAL_LR | 0.00001 |
| Final learning rate for $\beta$ | G_FINAL_LR | 0.00005 |
| Optimiser for both networks | (-) | Adam |
| EBM adam opt. beta 1 | E_BETA_1 | 0.5 |
| EBM adam opt. beta 2 | E_BETA_2 | 0.999 |
| GEN adam opt. beta 1 | G_BETA_1 | 0.5 |
| GEN adam opt. beta 2 | G_BETA_2 | 0.999 |
| LR scheduler | (-) | Exponential decay |
| LR scheduler start | BEGIN_EPOCH | 1 |
| LR scheduler decay | DECAY_RATE | 0.975 |
| LR schedule step | STEP_INTERVAL | 1 |
| MCMC sampler | (-) | Unadjusted Langevin |
| $s$ for $p_\alpha(\mathbf{z})$ (in Eq. 9) | E_STEP_SIZE | 0.16 |
| $s$ for $p_\theta(\mathbf{z}\|\mathbf{x}, t)$ (in Eq. 9) | G_STEP_SIZE | 0.01 |
| $K_{\mathbf{z}}$ for $p_\alpha(\mathbf{z})$ (in Eq. 9) | E_SAMPLE_STEPS | 60 |
| $K_{\mathbf{z}\|\mathbf{x},t}$ for $p_\theta(\mathbf{z}\|\mathbf{x}, t)$ (in Eq. 9) | G_SAMPLE_STEPS | 20 |
| $p$ (in Eq. 26) | TEMP_POWER | 0 for vanilla model, variable otherwise |
| $N_t$ (in Eq. 23) | NUM_TEMPS | 10 |
| $D_{\text{KL}}(\cdot\|\|\cdot)$ weight, $\eta$ in Eq. 27 | KL_BIAS_WEIGHT | 1 |

Table 4: Hyperparamter setup for CelebA

| Param. | Codebase Var. Name | Value |
|---|---|---|
| Dataset | DATASET | CIFAR10 |
| Number of training epochs | NUM_EPOCHS | 50 |
| Batch size | BATCH_SIZE | 75 (variable for vanilla model) |
| Number of training examples | NUM_TRAIN_DATA | 12000 |
| Number of testing examples | NUM_VAL_DATA | 4500 |
| Number of repetitions to conduct | NUM_EXPERIMENTS | 5 |
| Latent space dimension | Z_CHANNELS | 100 |
| Complexity of EBM, ($nez$ in tab. 6) | EBM_FEATURE_DIM | 100 |
| EBM Leaky-ReLU leak coefficient | EBM_LEAK | 0.1 |
| Complexity of GEN, ($ngz$ in tab. 6) | GEN_FEATURE_DIM | 64 |
| GEN Leaky-ReLU leak coefficient | GEN_LEAK | 0.2 |
| $\sigma_0$ (for $\pi_0(\mathbf{z})$ in Eq. 1) | p0_SIGMA | 1 |
| $\sigma_l$ (for $\epsilon$ in eq 2) | LKHOOD_SIGMA | 0.3 |
| Initial learning rate for $\alpha$ | E_INITIAL_LR | 0.00002 |
| Initial learning rate for $\beta$ | G_INITIAL_LR | 0.0001 |
| Final learning rate for $\alpha$ | E_FINAL_LR | 0.00001 |
| Final learning rate for $\beta$ | G_FINAL_LR | 0.00005 |
| Optimiser for both networks | (-) | Adam |
| EBM adam opt. beta 1 | E_BETA_1 | 0.5 |
| EBM adam opt. beta 2 | E_BETA_2 | 0.999 |
| GEN adam opt. beta 1 | G_BETA_1 | 0.5 |
| GEN adam opt. beta 2 | G_BETA_2 | 0.999 |
| LR scheduler | (-) | Exponential decay |
| LR scheduler start | BEGIN_EPOCH | 1 |
| LR scheduler decay | DECAY_RATE | 0.975 |
| LR schedule step | STEP_INTERVAL | 1 |
| MCMC sampler | (-) | Unadjusted Langevin |
| $s$ for $p_\alpha(\mathbf{z})$ (in Eq. 9) | E_STEP_SIZE | 0.16 |
| $s$ for $p_\theta(\mathbf{z}\|\mathbf{x}, t)$ (in Eq. 9) | G_STEP_SIZE | 0.01 |
| $K_{\mathbf{z}}$ for $p_\alpha(\mathbf{z})$ (in Eq. 9) | E_SAMPLE_STEPS | 60 |
| $K_{\mathbf{z}\|\mathbf{x},t}$ for $p_\theta(\mathbf{z}\|\mathbf{x}, t)$ (in Eq. 9) | G_SAMPLE_STEPS | 40 |
| $p$ (in Eq. 26) | TEMP_POWER | 0 for vanilla model, variable otherwise |
| $N_t$ (in Eq. 23) | NUM_TEMPS | 10 |
| $D_{\mathrm{KL}}(\cdot\|\|\cdot)$ weight, $\eta$ in Eq. 27 | KL_BIAS_WEIGHT | 1 |

Table 5: Hyperparamter setup for CIFAR-10

### 8.1.2 Network architectures

| Model | Layers | In-Out Size, Stride |
|---|---|---|
| | Input: z | Z_CHANNELS |
| EBM | Linear, LReLU | $nez$ - |
| for both | Linear, LReLU | $nez$ - |
| | Linear | 1 - |
| | Input: z | 1x1xZ_CHANNELS |
| | 4x4 convT($ngf$ x 16), LReLU | 4x4x($ngf$ x 16), 1 |
| GEN for | 4x4 convT($ngf$ x 8), LReLU | 4x4x($ngf$ x 8), 2 |
| CIFAR-10 | 4x4 convT($ngf$ x 4), LReLU | 4x4x($ngf$ x 4), 2 |
| | 4x4 convT($ngf$ x 2), LReLU | 4x4x($ngf$ x 2), 2 |
| | 4x4 convT(3), Tanh | 4x4x3, 2 |
| | Input: z | 1x1xZ_CHANNELS |
| | 4x4 convT($ngf$ x 16), LReLU | 4x4x($ngf$ x 16), 1 |
| | 4x4 convT($ngf$ x 8), LReLU | 4x4x($ngf$ x 8), 2 |
| GEN for CelebA | 4x4 convT($ngf$ x 4), LReLU | 4x4x($ngf$ x 4), 2 |
| | 4x4 convT($ngf$ x 2), LReLU | 4x4x($ngf$ x 2), 2 |
| | 4x4 convT($ngf$ x 1), LReLU | 4x4x($ngf$ x 1), 2 |
| | 4x4 convT(3), Tanh | 4x4x3, 2 |

Table 6: Model architectures for different datasets. These have been significantly simplified compared to those in [16]. Unlike the U-Net architecture's decoder used for the generator in [16], our generator models maintain a consistent kernel size throughout. These adjustments do not impact the results and conclusions, which remain comparative across different models.

## 8.2 Risk assessment retrospective

Since this was a computer-based project without any physical components, there were essentially no major risks or hazards encountered during the course of the work.

Any potential hazards related to prolonged computer use, such as ergonomic risks or eye strain from extended screen time, were minor and easily mitigated through proper workplace setup and taking regular breaks.

## 8.3 Supplementary results

### 8.3.1 Computational cost

Presented below are the associated computational costs for the models trained on CelebA. Thermodynamic Integration, whilst powerful, requires substantially more compute time.

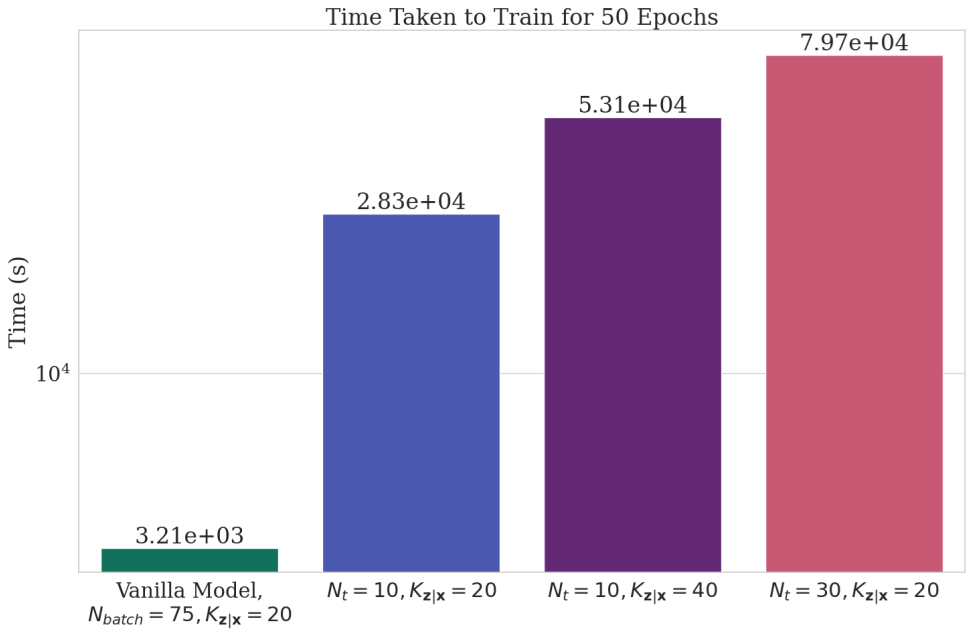



Figure 14: Training times for different models on CelebA over 50 epochs, with each training loop constrained to use the same GPU RAM.

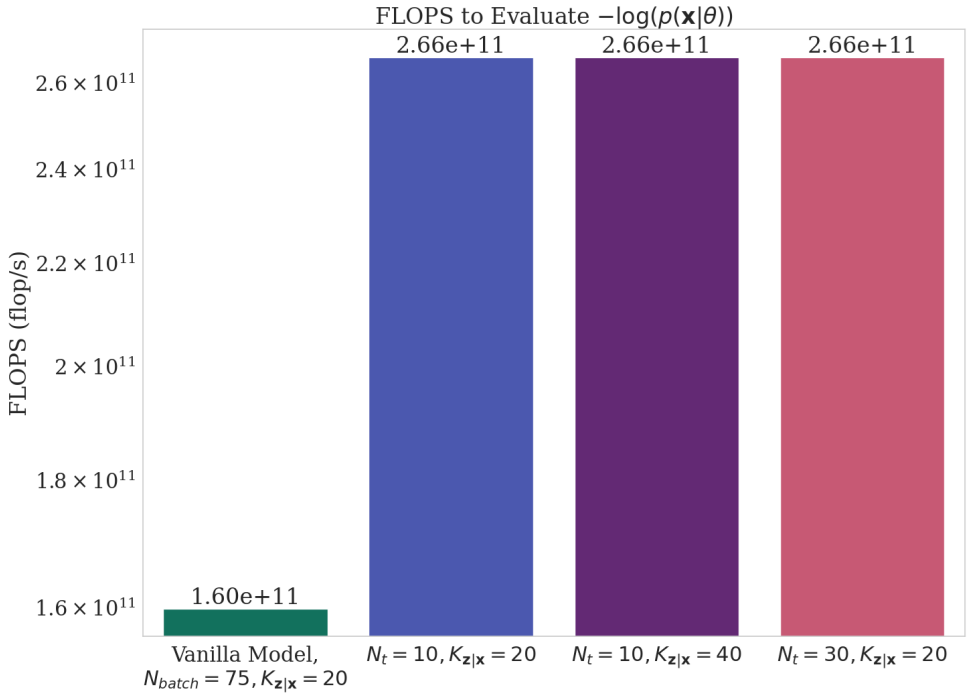



Figure 15: Floating point operations per second (FLOPs) required for the validation step, (i.e. loss evaluation). FLOPs were recorded using JAX's xla_computation function. Despite apparently similar FLOPs for models with $N_t = 10$ and $N_t = 30$, training times differ (Fig. 14). This discrepancy may be attributed to limitations regarding the estimates provided by the FLOPs profiler.

### 8.3.2 CIFAR-10 results

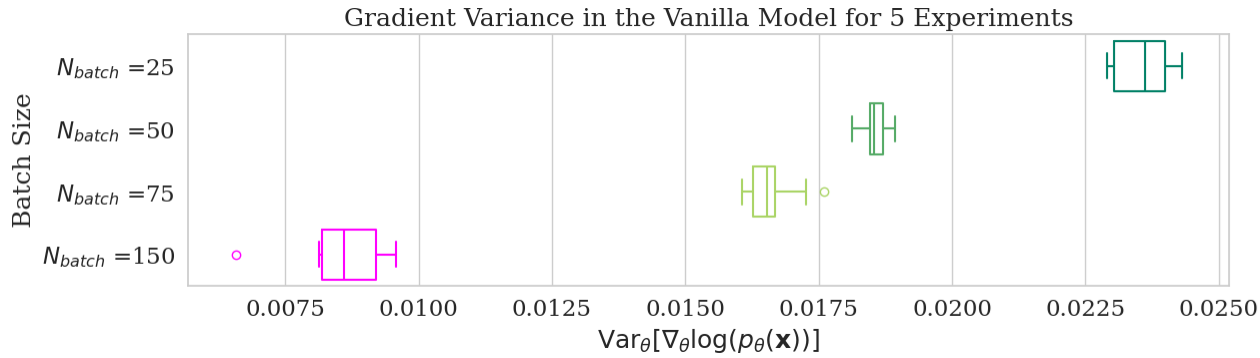



Figure 16: Relationship between final learning gradient variance of the vanilla model and batch size. The models were trained on CIFAR-10 for 50 epochs.

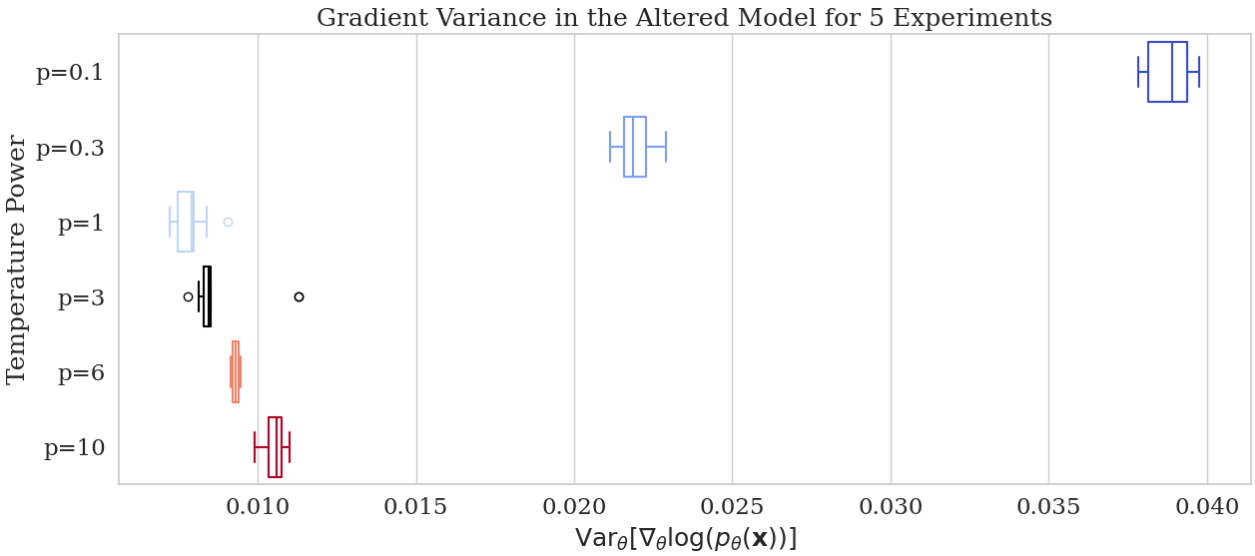



Figure 17: Relationship between final learning gradient variance and p in Eq. 26 the altered model. The models were once again trained on CIFAR-10 for 50 epochs. The same trends visible in Fig. 9 are apparent here.

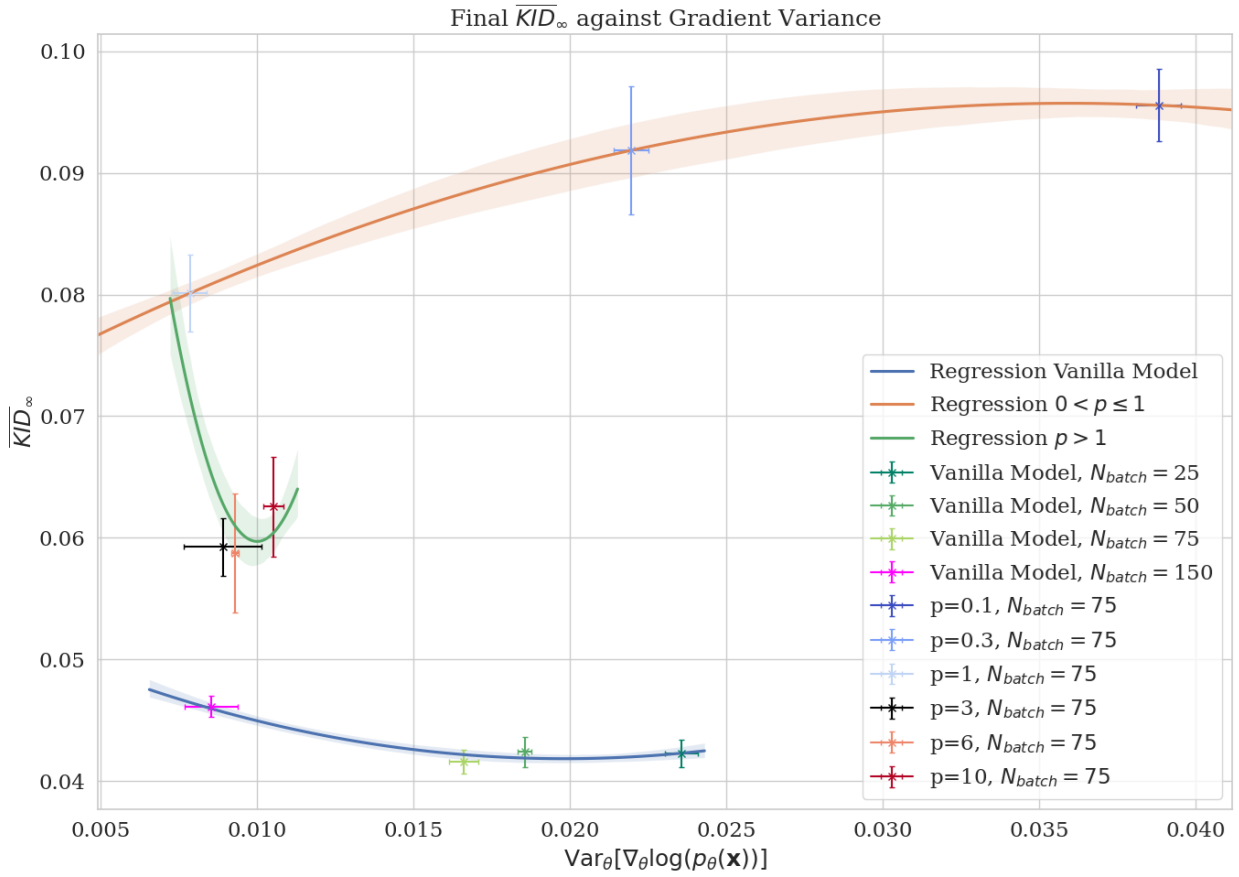



Figure 18: Learning gradient variance and $\overline{KID}_\infty$ for the models trained on CIFAR-10. The trends mirror that of Fig. 10, although the discrepancy between the vanilla and altered models has seemingly worsened and become more exaggerated.

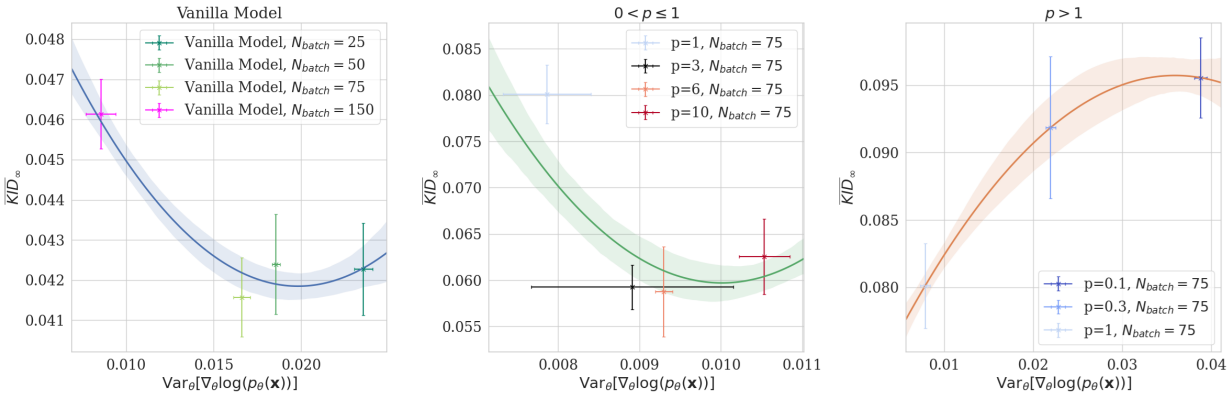



Figure 19: The relationships between learning gradient noise and generated image fidelity, as quantified by $\overline{KID}_\infty$ for CIFAR-10. This plot, equivalent to Fig. 18, has been separated into its different regimes for more clarity.

(a) Vanilla Model $N_{batch} = 75, K_{\mathbf{z}|\mathbf{x}} = 40$

(b) $p = 0.1, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 40$

(c) $p = 1, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 40$

(d) $p = 6, N_t = 10, K_{\mathbf{z}|\mathbf{x},t} = 40$

Figure 20: Final CIFAR-10 images generated by the models depicted in Fig. 18. The discrepancy described in Section 5 between the altered and vanilla models has become more pronounced in the case of CIFAR-10.

### 8.3.3 Train loss

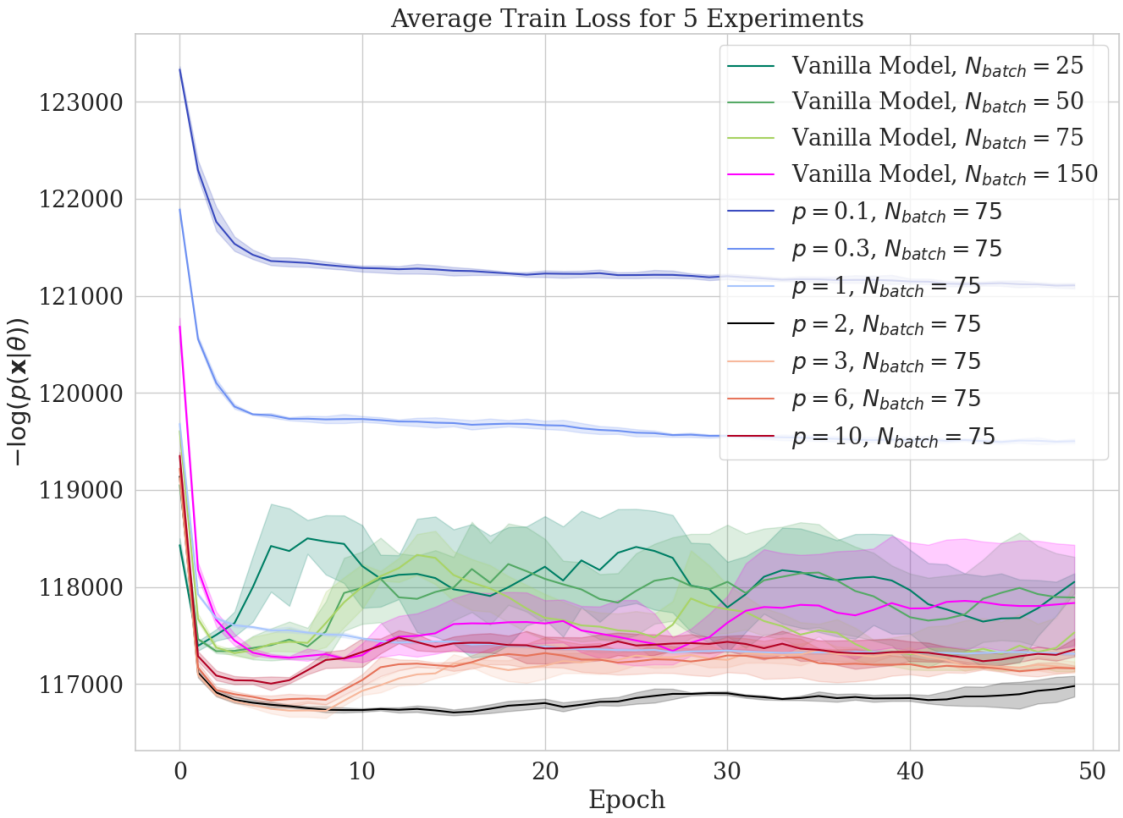



Figure 21: Evolution of training loss across five repetitions for each model trained on the CelebA dataset for 50 epochs.

The loss curves in Fig. 21 exhibit notable variability due to normalisation, which has been applied to enable a easier comparison between the vanilla and altered models.

The normalisation removes an inherent bias that accumulates throughout the thermodynamic integral, and offsets the models with unnormalised log-probability distributions from each other.

These offsets make it difficult to compare the models without normalisation. However, as shown, the normalisation results in variability in the loss curves, attributed to the discrete means used in the normalisation process, introducing associated errors similar to the discretised estimator of the mean error described in Eq. 19.

The most important illustration from the figure is that the models were generally able to converge within 50 epochs, and that $p = 0.1$ and $p = 0.3$ have seemingly converged to a local optima, suggesting mode-collapse.

### 8.3.4 $\overline{FID}_\infty$

As discussed in Section 4.4.3 $\overline{FID}_\infty$ was found to be a less reliable metric than $\overline{KID}_\infty$, when visually inspecting the generated images. However, the results incorporating $\overline{FID}_\infty$ have been included here for completeness. These are generally more sporadic and erroneous than the well-defined results presented in Section 4.2.

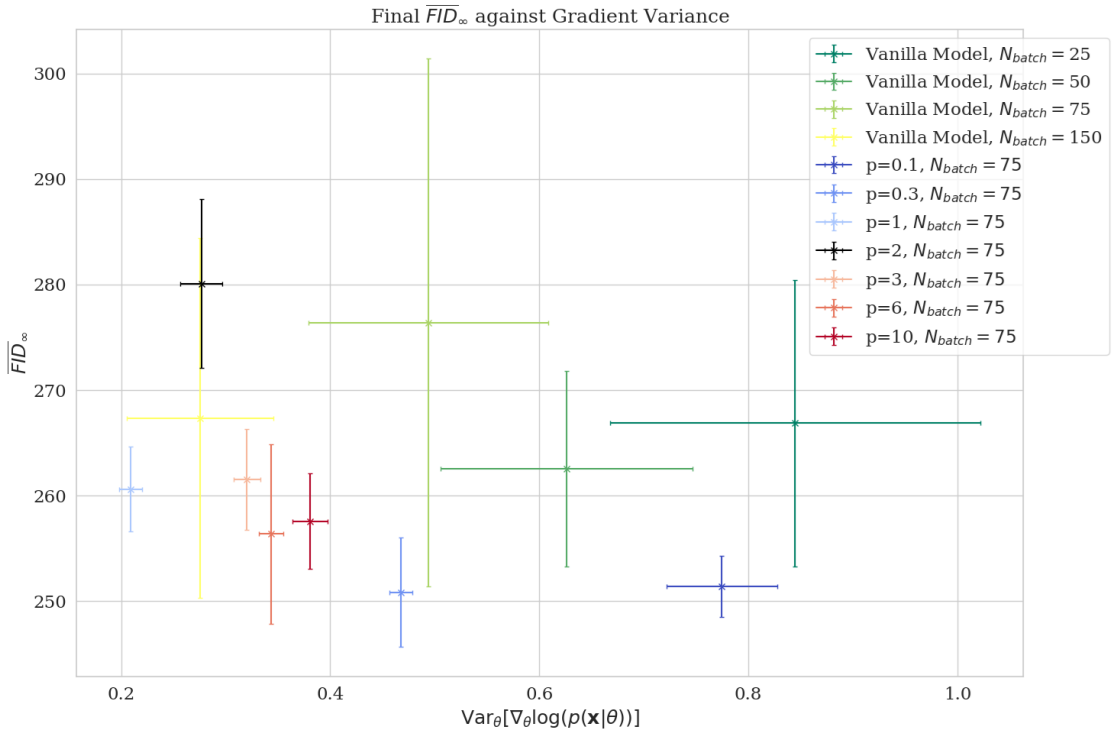



Figure 22: The effect of learning gradient variance on $\overline{FID}_\infty$ for the models trained on CelebA.

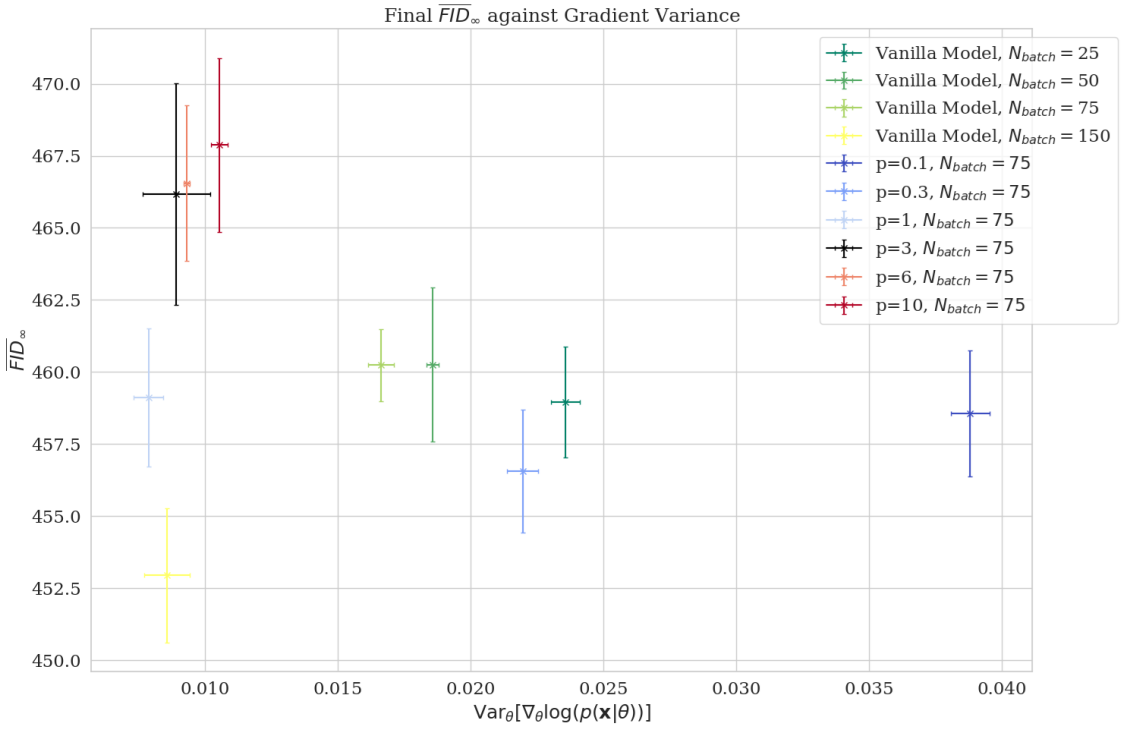



Figure 23: The effect of learning gradient variance on $\overline{FID}_\infty$ for the models trained on CIFAR-10.