Posterior Mean Matching: Generative Modeling through Online Bayesian Inference

Sebastian Salazar[1,3] **Michal Kucer**[1] **Yixin Wang**[2] **Emily Casleton**[1] **David Blei**[3]
[1]Los Alamos National Laboratory [2]University of Michigan [3]Columbia University

Abstract

This paper introduces posterior mean matching (PMM), a new method for generative modeling that is grounded in Bayesian inference. PMM uses conjugate pairs of distributions to model complex data of various modalities like images and text, offering a flexible alternative to existing methods like diffusion models. PMM models iteratively refine noisy approximations of the target distribution using updates from online Bayesian inference. PMM is flexible because its mechanics are based on general Bayesian models. We demonstrate this flexibility by developing specialized examples: a generative PMM model of real-valued data using the Normal-Normal model, a generative PMM model of count data using a Gamma-Poisson model, and a generative PMM model of discrete data using a Dirichlet-Categorical model. For the Normal-Normal PMM model, we establish a direct connection to diffusion models by showing that its continuous-time formulation converges to a stochastic differential equation (SDE). Additionally, for the Gamma-Poisson PMM, we derive a novel SDE driven by a Cox process, which is a significant departure from traditional Brownian motion-based generative models. PMMs achieve performance that is competitive with generative models for language modeling and image generation.

1 INTRODUCTION

The goal of generative modeling is to use data $\{\boldsymbol{x}_i\}_{i=1}^n$ to produce new samples from a target distribution $p^\star(\boldsymbol{x})$. The challenge is that $\boldsymbol{x}$ is high dimensional and $p^\star(\boldsymbol{x})$ is complex (MacKay, 2003).

Here are some examples:

- The data are natural images; the target is the distribution of images found in the world; the goal is to produce realistic images (Ho et al., 2020; Goodfellow et al., 2014).
- The data are documents; the target is the distribution of fluent language; the goal is to produce coherent text (Vaswani, 2017).
- The data are gene sequences of proteins; the target is the distribution of stable proteins; the goal is to produce new proteins with specific properties (Watson et al., 2023).
- Probabilistic prediction in tabular data, where the goal is to model the conditional distribution of a response variable given a collection of features (Beltran-Velez et al., 2024; Salazar, 2024).

In this paper, we develop *posterior mean matching* (PMM), a new method of generative modeling that is flexible enough to solve all of these problems. The key property of PMM is that it is based on the machinery of online Bayesian inference. It inherits the flexibility of Bayesian modeling, and so it is easy to apply to many types of data and target distributions.

To develop PMM, we first posit a conjugate Bayesian model and show how, in theory, it can be used to sample exactly from the target $p^*(\boldsymbol{x})$. We then show how to use variational inference variational inference to approximate a distribution that produces such exact samples. PMM is flexible because it can employ any conjugate Bayesian model in an inner routine.

We study PMM on images and text. For image generation, we develop a PMM method based on an underlying Gaussian/Gaussian model. We find that it pro-

For further correspondence, please email {ssalazar, michal}@lanl.gov.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

duces Frechet inception distance (FID) scores (Heusel et al., 2017) that are comparable to most diffusion models (Karras et al., 2022; Dhariwal and Nichol, 2021; Song et al., 2020). To apply PMM to text, we simply swap the Gaussian model for a Dirichlet/-Categorical. We find that the text-generating PMM models offer competitive performance to diffusion non-autoregressive language models (Lou et al., 2024; Sahoo et al., 2024a; Shi et al., 2024).

Related Work. Generative modeling is an active area of machine learning research. For images, some popular methods include variational autoencoders (Kingma, 2013; Rezende et al., 2014), generative adversarial networks (Goodfellow et al., 2014), normalizing flows (Dinh et al., 2014; Rezende and Mohamed, 2015), autoregressive models (Van den Oord et al., 2016), and diffusion models (Ho et al., 2020). For text, the main method is the transformer-based autoregressive models (Vaswani, 2017). PMM is a contribution to this research area, providing an easily adaptable method for generative modeling, applicable to text, images, and many other types of data. While on images PMM compares favorably to diffusions, on text, its performance is competitive with other non-autoregressive diffusion-based language models (Austin et al., 2023; Lou et al., 2024; Sahoo et al., 2024a; Shi et al., 2024). PMMs are also related to diffusion models, we establish this technical connection in Section 4.

Closest in spirit to PMMs is Bayesian flow nets (Graves et al., 2024) (BFNs), which also use Bayesian methods in the context of generative modeling. PMM is based on exact sampling from the target, while BFNs are motivated by information theoretic principles. PMM provides a simpler algorithm than BFN, and performed better in our studies of text data in Section 5.

Contribution. Posterior mean matching (PMM) contributes to the field of generative modeling by offering a unified and adaptable method grounded in Bayesian inference. PMM easily applies to diverse data types such as images, text, and count data.

2 POSTERIOR MEAN MATCHING

There are several ingredients to posterior mean matching. Throughout, we assume that we are given a dataset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of i.i.d. samples from the target distribution $p^*(\boldsymbol{x})$.

Noisy observation model. The first ingredient is the *noisy observation model.* It is a conditional distribution $\pi_\alpha(\boldsymbol{y}|\boldsymbol{x})$ that is easy to sample from (e.g., a Gaussian, Poisson, Categorical). Samples from this conditional are interpreted as noisy versions of $\boldsymbol{x}$.

Augmented Target Distribution. We augment the target distribution $p^*(\boldsymbol{x})$ with the noisy observation model $\pi_{\alpha_s}(\boldsymbol{y} \mid \boldsymbol{x})$ and define a joint distribution over $(\boldsymbol{x}, \boldsymbol{y}_{1:t})$, termed the *augmented target distribution*:

$$\boldsymbol{x} \sim p^*(\boldsymbol{x}), \tag{1}$$

$$y_s \mid \boldsymbol{x} \overset{\perp}{\sim} \pi_{\alpha_s}(\boldsymbol{y} \mid \boldsymbol{x}), \quad s = 1, \ldots, t, \tag{2}$$

$$p(\boldsymbol{x}, \boldsymbol{y}_{1:t}) \equiv p^*(\boldsymbol{x}) \prod_{s=1}^{t} \pi_{\alpha_s}(\boldsymbol{y}_s \mid \boldsymbol{x}). \tag{3}$$

Where we have introduced a sequence of hyperparameters $\alpha_1, ..., \alpha_t$, where α_s can be interpreted as a parameter modulating the amount of noise in the sample $\boldsymbol{y}_s$ (e.g., the precision parameter of a Normal distribution). The augmented model simply augments draws $\boldsymbol{x}^*$ from the target $p^*(\boldsymbol{x})$ with a collection of noisy observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$. Note the "prior" distribution (1) of this generative process is the target distribution $p^*(\boldsymbol{x})$, which is not directly available.

Augmented Bayesian Model. The next ingredient is the *augmented Bayesian model.* This model is identical to the *augmented target distribution* except that the unknown target $p^*(\boldsymbol{x})$ is replaced with a known distribution $\pi(\boldsymbol{x})$, that serves as a known "prior." The augmented Bayesian model is

$$\boldsymbol{x} \sim \pi(\boldsymbol{x}), \tag{4}$$

$$\boldsymbol{y}_s \mid \boldsymbol{x} \overset{\perp}{\sim} \pi_{\alpha_s}(\boldsymbol{y} \mid \boldsymbol{x}), \quad s = 1, \ldots, t \tag{5}$$

$$\pi(\boldsymbol{x}, \boldsymbol{y}_{1:t}) \equiv \pi(\boldsymbol{x}) \prod_{s=1}^{t} \pi_{\alpha_s}(\boldsymbol{y}_s \mid \boldsymbol{x}). \tag{6}$$

We require the augmented Bayesian model to satisfy the following three properties.

First, the posterior expectation $\mathbb{E}_\pi(\boldsymbol{x}|\boldsymbol{y}_{1:t})$ must have a known closed form. This is facilitated by picking a prior $\pi(\boldsymbol{x})$ that is conjugate to the noise model $\pi_{\alpha_s}(\boldsymbol{y}|\boldsymbol{x})$, e.g., a normal prior with a normal noisy observation model.

Second, the posterior mean must be *consistent.* Given a collection of noisy samples $\boldsymbol{y}_s \sim \pi_{\alpha_s}(\boldsymbol{y}|\boldsymbol{x}^*)$ for $s = 1, ..., t$, we say that the posterior mean is consistent if it eventually recovers the *true* $\boldsymbol{x}^*$ that was used to generate the noisy samples $\boldsymbol{y}_1, ..., \boldsymbol{y}_t$, namely

$$\boldsymbol{\mu}_t \equiv \mathbb{E}_\pi(\boldsymbol{x}|\boldsymbol{y}_{1:t}) \overset{a.s.}{\to} \boldsymbol{x}^*. \tag{7}$$

All of the PMM models considered in this paper are consistent (see Appendix for consistency proofs).

Finally, the augmented model must be amenable to online Bayesian inference. This means that it is possible

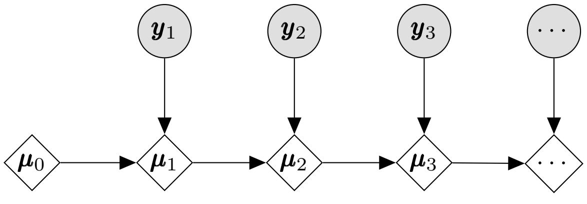


Figure 1: Diagram of the online Bayesian inference update process. At each time step t, an observation $\boldsymbol{y}_t$ is incorporated to update the posterior mean $\boldsymbol{\mu}_t$. The ellipsis $(\cdots)$ indicates the iterative nature of the updates, starting from the prior mean $\boldsymbol{\mu}_0$.

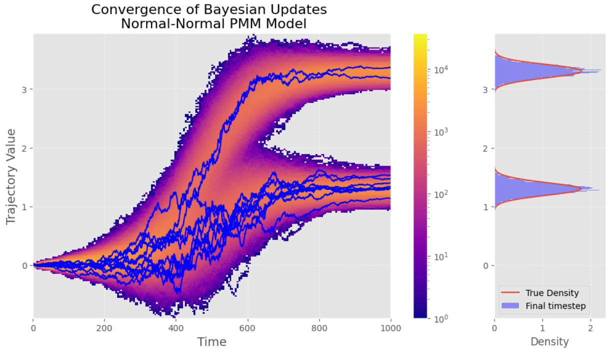


Figure 2: Convergence of the posterior mean trajectories $\boldsymbol{\mu}_t$ to samples from the target $\boldsymbol{x} \sim p^*(\boldsymbol{x})$ as t increases for the Normal Posterior Mean Matching (PMM) model. Refer to Figure C.1 in the Appendix C for a more detailed view.

to write an update rule for the posterior mean

$$\boldsymbol{\mu}_{t+1} = f_t(\boldsymbol{\mu}_t, \boldsymbol{y}_{t+1}). \tag{8}$$

Figure 1 diagrams online Bayesian inference.

Generative Modeling with online Bayesian Inference. With these ingredients—the augmented target and the augmented model—we show how to use augmented data from (3) and online Bayesian inference from the augmented Bayesian model (6) to produce a neural-network-based sampler from the target distribution $p^*(\boldsymbol{x})$.

We start by considering data from the augmented target distribution $\{\boldsymbol{x}^*, \boldsymbol{y}_1, ..., \boldsymbol{y}_t\}$, which we generate by taking a sample $\boldsymbol{x}^*$ from the target distribution $p^*(\boldsymbol{x})$ and then producing a sequence of t noisy observations $\boldsymbol{y}_{1:t}$ using the noise model $\pi_{\alpha_t}(\boldsymbol{y}|\boldsymbol{x})$. In practice, we approximate the target distribution $p^*(\boldsymbol{x})$ by taking a random sample $\boldsymbol{x}_i$ from our dataset $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ of i.i.d. samples from $p^*(\boldsymbol{x})$.

Using the augmented sample $\{\boldsymbol{x}^*, \boldsymbol{y}_1, ..., \boldsymbol{y}_t\}$, we consider the sequence of posterior means $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t$ with respect to the augmented Bayesian model; the sequence of posterior means has the following properties:

- $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t$ is a sequence of random variables. Their randomness is inherited from the noisy observation model $\pi_{\alpha_t}(\boldsymbol{y}|\boldsymbol{x})$.
- For all $s \in \{0, ..., t-1\}$, it is easy to calculate $\boldsymbol{\mu}_{s+1}$ from $\boldsymbol{\mu}_s$ and $\boldsymbol{y}_{s+1}$ using online Bayesian inference. This was one of the requirements of the augmented Bayesian model.
- The limit $\lim_{t\to\infty} \boldsymbol{\mu}_t$ converges to $\boldsymbol{x}^*$ — a sample from the target distribution $p^*(\boldsymbol{x})$. This is a consequence of the consistency of the posterior expectation, another requirement of the augmented Bayesian model.

These three properties suggest a strategy to draw samples from the target $p^*(\boldsymbol{x})$.

1. Obtain a sample $\{\boldsymbol{x}^*, \boldsymbol{y}_1, ..., \boldsymbol{y}_t\}$ and throw away $\boldsymbol{x}^*$. This results in a sequence $\boldsymbol{y}_1, ..., \boldsymbol{y}_t$, that is viewed as a sample from the marginal distribution of the augmented target (3).
2. Using the augmented Bayesian model, compute the sequence of posterior means $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t$ using online Bayesian Inference. It is worth highlighting that the expectation is defined using the augmented Bayesian model of equation (6), while the data $\boldsymbol{y}_{1:t}$ used to compute this expectation, are random variables drawn from the marginal distribution of the augmented target (3).
3. Because this sequence is consistent, for t large enough $\boldsymbol{\mu}_t \approx \boldsymbol{x}^*$. In other words, the posterior mean $\boldsymbol{\mu}_t$ is effectively a sample from the target distribution $p^*(\boldsymbol{x})$.

This logic implies that sampling from the target distribution $p^*(\boldsymbol{x})$ reduces to sampling $\boldsymbol{\mu}_t$ from the joint distribution of posterior means $p(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t)$.

We illustrate sample trajectories $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t$ in Figure 2, where the target is a bimodal distribution. We can see that $\boldsymbol{\mu}_t$ converges to samples from the target $p^*(\boldsymbol{x})$.

Approximately Sampling from the Target. In practice, the joint distribution $p(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_t)$ is intractable to sample from exactly. So, we sample from the target by approximating the joint distribution of posterior means, and taking samples of $\boldsymbol{\mu}_t$.

We approximate $p(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_t)$ by introducing a family of distributions $q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_t)$ and minimizing the following objective function:

$$\mathcal{L}_{\mathrm{PMM}}(\boldsymbol{\varphi}) = \mathrm{KL}(p(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_t) \| q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_t)). \tag{9}$$

The form of $q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{1:t})$ is motivated by mechanics of on-

Algorithm 1 Sampling $p^*(\boldsymbol{x})$ from a fitted PMM

```
Initialize: Set $\boldsymbol{\mu}_0$ to the prior mean.
for $s = 1$ to $t$ do
    Compute $\hat{\boldsymbol{x}}_s \leftarrow g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, s)$
    Sample $\hat{\boldsymbol{y}}_s \sim \pi_{\alpha_s}(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_s)$
    Update $\boldsymbol{\mu}_s$ using the online Bayesian Inference update rule $\boldsymbol{\mu}_s = f_s(\boldsymbol{\mu}_{s-1}, \hat{\boldsymbol{y}}_s)$
end for
Output: Return $\boldsymbol{\mu}_t$
```

line Bayesian inference and is defined implicitly:

$$\hat{\boldsymbol{y}}_{t+1} \sim \pi_{\alpha_{t+1}}(\boldsymbol{y} \mid g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_t, t)) \tag{10}$$

$$\boldsymbol{\mu}_{t+1} = f_t(\boldsymbol{\mu}_t, \hat{\boldsymbol{y}}_{t+1}). \tag{11}$$

Here, $g_{\boldsymbol{\varphi}}$ is a flexible function parameterized with a neural network. Given data, we learn the neural network $g_{\boldsymbol{\varphi}}$ by minimizing the PMM objective in Equation (9). Once fit, we can obtain approximate samples from the target distribution $p^*(\boldsymbol{x})$ by iteratively applying equations (10) and (11). This sampling procedure is detailed in Algorithm 1.

3 EXAMPLES OF PMM MODELS

We now work out the components of posterior mean matching using three conjugate pairs of distributions: Normal-Normal, Gamma-Poisson, and Dirichlet-Categorical models. These models are suitable for real-valued, positive, and text data, respectively.

3.1 Normal-Normal PMM: a generative model of real-valued data

Data Representation. This section concerns the Normal-Normal PMM, a generative model designed to model real-valued data. This boils down to assuming that samples from the target distribution $p^*(\boldsymbol{x})$ are vectors in $\mathbb{R}^d$.

Augmented Target Distribution. The Normal-Normal PMM posits a *noisy observation model* that corrupts samples $\boldsymbol{x}^*$ from the target distribution $p^*(\boldsymbol{x})$ through additive Gaussian noise $\boldsymbol{y}_t \sim \mathcal{N}(\boldsymbol{x}^*, \alpha_t^{-1} I)$. This noisy observation model defines the following augmented target distribution

$$p(\boldsymbol{x}, \boldsymbol{y}_{1:t}) \equiv p^*(\boldsymbol{x}) \prod_s \mathcal{N}(\boldsymbol{y}_s; \boldsymbol{x}, \alpha_s^{-1} I) \tag{12}$$

In this context, the precision parameter α_t modulates the level of corruption in the noisy observations.

Augmented Bayesian Model. Suppose we are given a sample $\boldsymbol{y}_{1:t}$ from the marginal distribution of the augmented target

$$p(\boldsymbol{y}_{1:t}) = \int p^*(\boldsymbol{x}) \prod_s \mathcal{N}(\boldsymbol{y}_s; \boldsymbol{x}, \alpha_s^{-1} I) d\boldsymbol{x}. \tag{13}$$

Based on equations (12) and (13) we know that there exists an $\boldsymbol{x}^*$, that is a sample from $p^*(\boldsymbol{x})$ such that $\boldsymbol{y}_s \sim \mathcal{N}(\boldsymbol{x}^*, \alpha_s^{-1} I)$. However, we only assume that $\boldsymbol{y}_{1:t}$ is given to us—the mean parameter $\boldsymbol{x}^*$ of these normal distributions is kept hidden. We infer $\boldsymbol{x}^*$ by using a Normal-Normal augmented Bayesian model

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \beta^{-1} I) \tag{14}$$

$$\boldsymbol{y}_s | \boldsymbol{x}, \alpha_s^{-1} \sim \mathcal{N}(\boldsymbol{x}, \alpha_s^{-1} I). \tag{15}$$

Online Bayesian Inference Update. It is possible to calculate the posterior mean in the Normal-Normal model using the following update rule (see Appendix A.2.1)

$$\boldsymbol{\mu}_t | \boldsymbol{\mu}_{t-1}, \boldsymbol{y}_t = \frac{\beta + \sum_{s=1}^{t-1} \alpha_s}{\beta + \sum_{s=1}^{t} \alpha_s} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t \boldsymbol{y}_t}{\beta + \sum_{s=1}^{t} \alpha_s} \tag{16}$$

where $\boldsymbol{y}_t \sim \mathcal{N}(\boldsymbol{x}, \alpha_t^{-1} I)$. The following theorem rigorously establishes the convergence of this posterior mean to $\boldsymbol{x}^*$.

Theorem 1. *(Concentration of posterior mean) Let* $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ *be observations generated according to equation (13). Suppose* α_t *a known, positive, increasing sequence satisfying* $\lim_{t \to \infty} \alpha_t = \infty$. *Then, the posterior mean* $\boldsymbol{\mu}_t$ *of the Bayesian model in equations (14) and (15) is consistent, namely:*

$$\lim_{t \to \infty} \boldsymbol{\mu}_t = \boldsymbol{x}, \quad \textit{almost surely}, \tag{17}$$

with respect to the joint distribution of $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots)$ *in equation (12).*

Theorem 1 establishes the correctness of the approximate sampling scheme shown in Algorithm 1 for the Normal-Normal model, which implies that in the limit, the posterior mean of this Bayesian model is effectively a sample from the target $p^*(\boldsymbol{x})$. A visual demonstration of Theorem 1 is shown in Figure 2.

Putting together all of these components we now show how to compute the PMM objective for this model.

Normal-Normal Posterior Mean Matching Objective. Using the online Bayesian Inference update, we approximate the posterior mean updates of equation (16) using a Neural Network $g_{\boldsymbol{\varphi}}$ as in equations (10) and (11) as follows

$$\hat{\boldsymbol{y}}_s \sim \mathcal{N}(g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, s), \alpha_s^{-1}) \tag{18}$$

$$\boldsymbol{\mu}_t | \boldsymbol{\mu}_{t-1} = \frac{\beta + \sum_{s=1}^{t-1} \alpha_s}{\beta + \sum_{s=1}^{t} \alpha_s} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t \hat{\boldsymbol{y}}_t}{\beta + \sum_{s=1}^{t} \alpha_s} \tag{19}$$

Substituting (16) and (19) into the PMM objective we obtain (see appendix A.2.3)

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) \propto t \cdot \mathbb{E}_{\substack{s\sim U(\{1,\ldots,t\}) \\ \boldsymbol{x}\sim p^*(\boldsymbol{x}) \\ \boldsymbol{\mu}_{s-1}|\boldsymbol{x}}} \alpha_s \|\boldsymbol{x} - g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, s)\|_2^2 \tag{20}$$

3.2 Dirichlet-Categorical PMM: a generative model of text

Data Representation. In this section, we develop a Dirichlet-Categorical PMM to model a collection of text documents. This boils down to assuming that samples from the target distribution $p^*(\boldsymbol{X})$ come from a discrete, finite space. Specifically, we represent each document in a corpus as a sequence of tokens $\mathbf{X}^* = (\mathbf{x}_1, \ldots, \mathbf{x}_C)$, where each token $\mathbf{x}_c \in \{0,1\}^V \cap \Delta^{V-1}$ is one-hot encoded from a fixed vocabulary of size V.

Augmented Target Distribution. For every document $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_C)$, suppose we generate a sequence of noisy documents $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_T$ according to the following generative process:[1]

$$(\boldsymbol{x}_1, ..., \boldsymbol{x}_C) \sim p^*(\boldsymbol{x}_1, ..., \boldsymbol{x}_C), \tag{21}$$

$$\boldsymbol{y}_{tc}|\boldsymbol{x}_c, \alpha_{tc} \sim \text{Cat}_{V+1}\left(\alpha_{tc}x_c^{(1)}, ..., \alpha_{tc}x_c^{(V)}, 1-\alpha_{tc}\right). \tag{22}$$

Here, the $V+1^{th}$ token of the categorical random variable in equation (22) should be thought of as a <mask> token, representing missing data in the noisy observations of a document, and $w_{tc} \in [0,1]$ represents the probability that a token at position c is unmasked at time t.

Augmented Bayesian Model. A sample from the marginal distribution induced by augmented target distribution of equations (21) and (22) consists of a sequence of noisy versions $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_t$ of a document $\boldsymbol{X}$. It is possible to infer $\boldsymbol{X}$ with the marginal samples of this noisy observation model using the following augmented Bayesian model.

$$(\boldsymbol{x}_1, ..., \boldsymbol{x}_C) \sim_{\text{iid}} \text{Dir}_V(1/K), \tag{23}$$

$$\boldsymbol{y}_{tc}|\boldsymbol{x}_c, \alpha_{tc} \sim \text{Cat}_{V+1}\left(\alpha_{tc}x_c^{(1)}, ..., \alpha_{tc}x_c^{(V)}, 1-\alpha_{tc}\right). \tag{24}$$

Online Bayesian Inference: Encoding Prior Knowledge with a Non-Informative Prior. We present the online Bayesian update rule assuming a non-informative Dirichlet Prior (i.e. taking $K \to \infty$ in (23)). The purpose of this choice is twofold:

1. A non-informative Dirichlet prior pushes mass towards the vertices of the probability simplex $\{0,1\}^V \cap \Delta^{V-1}$; since we know that documents are represented by vertices of the probability simplex, this effectively encodes prior knowledge about the generative process directly into the noisy observation model. Encoding prior knowledge about the data this way is a noticeable advantage of PMMs that is not present in other generative modeling frameworks.
2. It simplifies the posterior mean dynamics which are given by (more details in section A.3.1 of the appendix)

$$\boldsymbol{\mu}_{sc}|(\boldsymbol{\mu}_{s-1,c} = \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) = \begin{cases} \boldsymbol{y}_{sc}^{(1:V)}, & \text{if } \boldsymbol{y}_{sc}^{(1:V)} \neq \mathbf{0} \\ \frac{\mathbb{1}_V}{V}, & \text{if } \boldsymbol{y}_{sc}^{(1:V)} = \mathbf{0} \end{cases} \tag{25}$$

$$\overset{d}{=} \begin{cases} \text{Cat}(\boldsymbol{x}_c) & \text{w/prob } \alpha_{tc} \\ \frac{\mathbb{1}_V}{V} & \text{w/prob } 1-\alpha_{tc} \end{cases} \tag{26}$$

$$\boldsymbol{\mu}_{tc}|(\boldsymbol{\mu}_{t-1,c} \neq \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) = \boldsymbol{\mu}_{t-1,c}. \tag{27}$$

We use these updates to derive the PMM objective.

Dirichlet-Categorical Posterior Mean Matching Objective. As before, we approximate the online Bayesian inference updates from equations (26) and (27), using a neural network.

$$\hat{\boldsymbol{x}}_{tc} = g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t) \tag{28}$$

$$\hat{\boldsymbol{y}}_{sc} \sim \text{Cat}_{V+1}(\alpha_{sc}\hat{x}_c^{(1)}, ..., \alpha_{tc}\hat{x}_c^{(V)}, 1-\alpha_{tc}) \tag{29}$$

$$\boldsymbol{\mu}_{sc}|(\boldsymbol{\mu}_{s-1,c} = \frac{\mathbb{1}_V}{V}) = \begin{cases} \text{Cat}(\hat{\boldsymbol{x}}_{tc}) & \text{w/prob } \alpha_{tc} \\ \frac{\mathbb{1}_V}{V} & \text{w/prob } 1-\alpha_{tc} \end{cases}. \tag{30}$$

Substituting (26) and (30) into the PMM objective, we obtain (see appendix A.3.1)

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) \propto$$

$$-\sum_c t\mathbb{E}_{\substack{s\sim U(\{1,\ldots,t\}) \\ \boldsymbol{x}\sim p^*(\boldsymbol{x}) \\ \boldsymbol{\mu}_{s-1}|\boldsymbol{x}}} \mathbb{1}\left(\boldsymbol{\mu}_{t-1,c} = \frac{\mathbb{1}_V}{V}\right) \alpha_{tc} \log g_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{t-1})_c \tag{31}$$

A continuous-time PMM objective for the Dirichlet Categorical Model. It is relatively straightforward to generalize the PMM objective of the Dirichlet-Categorical model to a continuous-time formulation. This generalization is obtained by taking the continuum limit. We defer the technical details of this formulation to Appendix A.3.4).

[1] Where $\boldsymbol{Y}_t = (\boldsymbol{y}_{t1}, ..., \boldsymbol{y}_{tC})$ with $\boldsymbol{y}_{tc} \in \{0,1\}^{V+1} \cap \Delta^V$

3.3 Posterior Mean Matching with Other Conjugate Pairs

In general, it is possible to apply the same logic we used to derive the Normal-Normal and Dirichlet-Categorical PMMs to other conjugate Bayesian models. Conjugacy is a powerful tool since it allows us to compute the posterior mean of a Bayesian model in closed form. We believe that generalizing Posterior Mean Matching to situations where the posterior mean is not available in closed form represents an exciting avenue for future work. Now that we are equipped with all of the tools necessary to derive PMM models, we briefly state the components of a Gamma-Poisson PMM below and defer the development of the Inverse Gamma-Gamma model to the Appendix.

Data Representation. The Gamma-Poisson model is suitable to model target distributions $p^*(\boldsymbol{x})$ of positive or count data.

Augmented Target Distribution. Given a sample from the target $\boldsymbol{x} \sim p^*(\boldsymbol{x})$, we consider the following noisy observation model $\boldsymbol{y}_t|\boldsymbol{x} \sim \text{Pois}(\alpha_t \boldsymbol{x})$. These components completely specify the augmented target distribution $p(\boldsymbol{y}_{1:t}) = \int p^*(\boldsymbol{x}) \prod_s \text{Pois}(\boldsymbol{y}_t; \alpha_t \boldsymbol{x}) d\boldsymbol{x}$.

Augmented Bayesian Model. Given a noisy sample $\boldsymbol{y}_{1:t}$ from the marginal distribution $p(\boldsymbol{y}_{1:t})$, it is possible to recover the sample from the target $\boldsymbol{x}^* \sim p^*(\boldsymbol{x})$ by using a Bayesian model with prior $\boldsymbol{x} \sim \Gamma(\beta_1, \beta_2)$ and likelihood $\boldsymbol{y}_t|\boldsymbol{x} \sim \text{Pois}(\alpha_t \boldsymbol{x})$.

Online Bayesian Inference Update. For the Gamma-Poisson PMM model the online Bayesian Inference update is given by

$$\boldsymbol{\mu}_s|\boldsymbol{\mu}_{s-1}, \boldsymbol{x} \overset{d}{=} \frac{\beta_2 + \sum_{k=1}^{s-1} \alpha_k}{\beta_2 + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\alpha_s \boldsymbol{y}_s}{\beta_2 + \sum_{k=1}^{s} \alpha_k}, \tag{32}$$

where $\boldsymbol{y}_s \sim \text{Pois}(\alpha_t \boldsymbol{x})$.

4 DIFFUSION MODELS AND SDEs

Theorem 1 establishes that the posterior mean $\boldsymbol{\mu}_t$ converges to the true observation $\boldsymbol{x}$ as more observations are incorporated. Intuitively, the iterative refinement of online Bayesian inference is analogous to the denoising steps of diffusion models, where each step incrementally reduces noise to approach the underlying data distribution. Here we formalize this intuition by mathematically connecting PMMs and stochastic differential equations (SDEs).

Specifically, the Bayesian update in the Normal-Normal PMM model can be interpreted as a discrete-time step in a type of diffusion process, with the posterior mean $\boldsymbol{\mu}_t$ acting as the denoising function steering towards the sample $\boldsymbol{x}$ from the target distribution $p^*(\boldsymbol{x})$. Although the continuous-time formulation of the Normal-Normal PMM is, strictly speaking, a diffusion process, we want to emphasize that the behavior and functional form of the SDEs are different from those typically appearing in the literature on diffusion models (Song et al., 2020). We connect the Normal-Normal PMM model to SDEs in the following theorem.

Theorem 2. ***(Online Bayesian Inference as a Diffusion Process)*** *Consider the update rule for the posterior mean $\boldsymbol{\mu}_t$ given by* (16)*. Let $f : [0, 1] \to \mathbb{R}^+$ be a monotonic function such that $\lim_{t \to 1} \int_0^t f(\tau) d\tau \to \infty$ and consider a partition of the unit interval $0 = t_1 < t_2 < \ldots < t_T = 1$. Moreover, define the sequence $\alpha_1, \ldots, \alpha_T$ in* (15) *by $\alpha_s = f(t_s)\delta t_s$. In the limit as $T \to \infty$ and $\delta t_s \to 0$, the discrete updates of* (16) *converge to a diffusion process defined by the following Stochastic Differential Equation (SDE):*

$$d\boldsymbol{\mu}(t) = f(t) \frac{(\boldsymbol{x} - \boldsymbol{\mu}(t))}{b + \int_0^t f(\tau)\, d\tau} dt + \frac{\sqrt{f(t)}}{b + \int_0^t f(\tau)\, d\tau} d\boldsymbol{W}_t, \tag{33}$$

$$\boldsymbol{x} \sim p^*(x). \tag{34}$$

What is surprising is that the continuous-time formulation of the Gamma-Poisson PMM model is also related to SDEs.

Theorem 3. *(Gamma-Poisson SDE) Consider the update rule of the posterior mean μ_t for the Gamma-Poisson PMM shown in equation (32). Let $f : [0, 1] \to \mathbb{R}^+$ and consider $0 = t_1 < t_2 < \ldots < t_T = 1$ a partition of the unit interval. Moreover, define the sequence $\alpha_1, \ldots, \alpha_T$ of the Gamma-Poison PMM by $\alpha_s = f(t_s)\delta t_s$. In the continuum limit $T \to \infty$ and $\max_s \delta t_s \to 0$, we have that the discrete updates of $\boldsymbol{\mu}_t$ converge to a Merton jump process characterized by the following Stochastic Differential Equation (SDE):*

$$d\boldsymbol{\mu}(t) = \left(L'(t) + \frac{A'(t)}{A(t)} (\boldsymbol{\mu}(t) - L(t)) \right) dt + A(t) d\boldsymbol{N}(t). \tag{35}$$

Where $\boldsymbol{N}(t)$ is a Cox Process with random base measure $\boldsymbol{x}dt$ with $\boldsymbol{x} \sim p^*(\boldsymbol{x})$, and $A(t) = (\beta_2 + \int_0^t f(\tau)d\tau)^{-1}$ and $L(t) = \beta_1(\beta_2 + \int_0^t f(\tau)d\tau)^{-1}$.

The proof is in the Appendix. Theorem 3 marks a significant departure from traditional Brownian motion-based generative models that typically appear in the literature on diffusion models (Song et al., 2020).

The Computation / Quality Trade-off. The connections between PMMs and SDEs established in Theorems 2 and 3 allow PMM models to use numerical techniques that have been developed to solve stochastic differential equations over many decades. This connection to SDEs allows us to interpret algorithm 1 as using the Euler-Maruyama method to numerically sample paths from an SDE. In the experiments of Section 5, we use this connection to SDEs to trade compute for sample quality.

5 EXPERIMENTS

We evaluate the performance of Posterior Mean Matching (PMM) models on image and text generation tasks. For image generation, we train Normal-Normal and Gamma-Poisson PMMs on three benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009), FFHQ-64 (Karras et al., 2019), and AFHQv2-64 (Choi et al., 2020). For our text experiments, we evaluate the performance of a Dirichlet-Categorical PMM on the text8 and OpenWebText dataset (Gokaslan and Cohen, 2019; Mahoney, 2011). The following is a summary of our findings:

- The Normal-Normal PMM achieves a competitive FID score of 2.18 on CIFAR-10, an FID score that is comparable to most diffusion models.
- The Gamma-Poisson PMM achieves an FID score of 4.36 on CIFAR-10. This score is lower than other diffusion models based on the Poisson likelihood (Chen and Zhou, 2023; Santos et al., 2023).
- Using the SDE interpretation of PMMs, we show that the FID scores of the Normal PMM degrade marginally when using a reduced number of function evaluations. Notably, on CIFAR-10 decreasing the number of function evaluations from 5000 to 166 (a factor of 30) reduces the FID score from 2.18 to 2.79.
- On OpenWebText the Dirichlet-Categorical PMM achieves a generative perplexity of 37.06 and 42.58 using top-350 and top-500 sampling, respectively, demonstrating performance on par with current non-autoregressive diffusion-based language models (Lou et al., 2024; Sahoo et al., 2024a; Shi et al., 2024).
- On text8, PMM achieves a bits per character (BPC) of 1.29, better than non-autoregressive language models based on diffusion. It narrows the gap to autoregressive language models, which achieve a BPC of 1.23.

5.1 Image Generation Tasks

Neural Network Architecture. In all of our experiments, we use an open-source implementation of the DDPM++ architecture (Karras et al., 2022; Dhariwal and Nichol, 2021).

Table 1: If FID scores are available for different sources, we report both scores (lower is better). The FID scores for these models may be found in Karras et al. (2022); Song et al. (2020). All of our experiments make use of class conditioning. The Normal-Normal PMM also uses adaptive data augmentation.

Method	**Cifar-10**	**FFHQ-64**	**AFHQv2-64**
DDPM	3.17	–	–
DDPM++	2.78	–	–
DDPM ++ (VP)	$2.18^*/2.55$	3.13^*	2.43^*
DDPM ++ (VE)	2.48^*	22.53^*	23.12^*
NSCN++ (VE)	2.38	–	–
NCSN ++ (VE, deep)	2.20	–	–
DDPM++ (VP, deep)	2.41	–	–
PFGM	2.48^*	–	–
PFGM, deep	2.35^*	–	–
Style GAN w/ADA	2.42	–	–
SOTA Diffusion	1.79	2.39	1.96
Normal PMM (ours)	**2.18**	3.41	2.48

NFE: Neural Function Evaluations.
(deep): Methods with this marker use deeper networks.
*: Indicates use of a higher order solver like RK-45 or Heun.

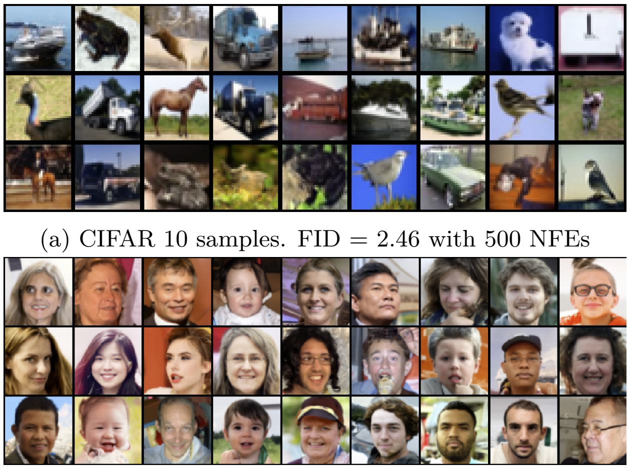
Table 2: FID scores for the Normal-Normal PMM as a function of neural function evaluations (NFEs).

Method / NFE	**100**	**166**	**500**	**1k**	**3k**	**5k**
Cifar10	3.98	2.79	2.46	2.33	2.28	2.18
AFHQ-v2	–	3.04	2.62	2.52	2.48	–
FFHQ	–	5.76	3.89	3.65	3.41	–

Discussion. The results of our experiments are shown in Table 1. Unless otherwise stated, the methods in Table 1 use very similar Neural Network architectures to the ones used in our experiments. A notable exception to this rule is Style-GAN (Karras et al., 2019). We report the performance of Style-GAN to paint a more complete picture of the performance of state-of-the-art models that are not based on diffusion.

We measure the quality of the generated images by computing the FID score on a sample of $50,000$ images. We also evaluate the Normal-Normal PMM on AFHQ-v2 and FFHQ-64, two higher-resolution datasets consisting of images of animals and humans, respectively. The performance of the Normal-Normal PMM on these datasets is also comparable to other popular diffusion models (see Table 1).

We compare the performance of the Gamma-Poisson PMM against other Poisson diffusion models. The Gamma-Poisson PMM achieves an FID score of 4.36, which is a better score than previous generative models that use the Poisson distribution. The details of the



(a) CIFAR 10 samples. FID = 2.46 with 500 NFEs

(b) FFHQ samples. FID = 3.89 with 500 NFEs

Figure 3: Comparison of sample generations for the Normal-Normal PMM model across CIFAR10 and FFHQ datasets. See Appendix C for a larger sample of generated images.

Gamma-Poisson model are shown in Appendix. In all of our image generation experiments, we choose an exponential schedule for the α_t's in the noise model (see Appendix B.1.2)

5.2 Language Modeling Tasks

For the text experiments, we fit a continuous-time Dirichlet-Categorical PMM model using a non-informative prior on two datasets: text8 and OpenWebText. The text8 dataset consists of the first 100M characters of cleaned English Wikipedia text, while OpenWebText is a large-scale corpus derived from web pages shared on Reddit with high engagement. On text8 we find that the Dirichlet-Categorical PMM outperforms all of the existing non-autoregressive baselines (see Table 3). Dirichlet-Categorical PMMs also achieve competitive Generative Perplexity on OpenWebText compared to other non-autoregressive (see Table 4).

Language Modeling We evaluate our Dirichlet-Categorical PMM against several baselines, including traditional autoregressive models and recent non-autoregressive approaches. On the text8 dataset, our model achieves a bits per character (BPC) of 1.29.

Unconditional Language Generation Following previous work (Lou et al., 2024; Sahoo et al., 2024a), we assess the quality of unconditional text output by computing generative perplexity relative to the `gpt2-large` language model (Radford et al., 2019). While generative perplexity is a standard metric for evaluating traditional autoregressive language models, its estimation for non-autoregressive models is more complex due to their inherently different generation processes. To ensure a fair comparison, we sample tokens using top-500 sampling and generate 1024 sequences of length 1024 from each model, using at most 1000 network evaluations for the non-autoregressive models. The resulting generative perplexities are shown in Table 4. where our model attains a competitive generative perplexity [2] with other non-autoregressive models.

Table 3: Bits per Character (BPC) Performance

Model	BPC
Autoregressive (GPT-2)	1.23
D3PM (Uniform)	1.61
D3PM (Absorb)	1.45
SEDD (Absorb)	1.39
Bayesian Flow Nets	1.41
GenMD4	1.34
Dirichlet-Categorical PMM (Ours)	**1.29**

Table 4: Generative Perplexity (**PPL**) measured at context length (CL) 1024 relative to GPT-2 large.

Model	PPL
AR	20.98
SEDD	30.96
MDLM	31.69
Dirichlet-Categorical PMM	42.58

While this is an exciting finding, it is important to note that evaluating unconditional non-autoregressive language models remains challenging, and it is known that benchmarks like generative perplexity are susceptible to manipulation through temperature annealing techniques (Lou et al., 2024). For this reason, we encourage readers to qualitatively assess the text generation capabilities of each model themselves by looking at the samples provided in the Appendix. These samples showcase our model's ability to generate coherent and diverse text across various topics and styles.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduced **Posterior Mean Matching (PMM)**, a novel and flexible framework for generative modeling grounded in Bayesian inference. PMM leverages conjugate pairs of distributions to model complex data distributions across various modalities, offering an alternative to traditional dif-

[2] PMM results were obtained using a different tokenizer, a smaller model with gpt2-based architecture, and without applying exponential moving averages (EMA).

fusion models. Through comprehensive experiments, we demonstrated the efficacy of PMM in both image and language generation tasks.

7 Ackowledgements

This work was conducted under the U.S. Department of Energy, National Nuclear Security Administration's Office of Defense Nuclear Nonproliferation Research and Development (NA-22) Steel Thread Venture. David Blei acknowledges support from the Simons Foundation, the National Science Foundation (grant numbers IIS-2127869 and DMS-2311108), and the Office of Naval Research (grant number N000142412243). Yixin Wang is supported by the National Science Foundation (grant numbers 2231174, 2310831, and 2428059), the Office of Naval Research (grant number N000142312590), and the Michigan Institute for Data Science Propelling Original Data Science (PODS) grant.

This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001.

We are deeply grateful to members of the Blei Lab for their invaluable contributions: Alessandro Grande for his thoughtful feedback and discussions on the early drafts, and Nicholas Beltran for reviewing an early version of the paper. We also thank Luhuan Wu, Sweta Karlekar, and Eli Weinstein for their insightful discussions during the initial stages of the project. We extend our appreciation to our colleagues at Los Alamos National Laboratory (LANL), Dr. Yen Ting Lin and Nihar Mauskar, for their thoughtful input, and to Carolyn Connor for her assistance in accessing the Venado computing cluster.

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. (2023). Structured denoising diffusion models in discrete state-spaces.

Beltran-Velez, N., Grande, A. A., Nazaret, A., Kucukelbir, A., and Blei, D. (2024). Treeffuser: Probabilistic predictions via conditional diffusions with gradient-boosted trees.

Chen, T. and Zhou, M. (2023). Learning to jump: Thinning and thickening latent counts for generative modeling. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 5367–5382. PMLR.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis.

Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Gokaslan, A. and Cohen, V. (2019). Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *NeurIPS*, 27.

Graves, A., Srivastava, R. K., Atkinson, T., and Gomez, F. (2024). Bayesian flow networks.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *NeurIPS*.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks.

Kingma, D. P. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lou, A., Meng, C., and Ermon, S. (2024). Discrete diffusion modeling by estimating the ratios of the data distribution.

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK.

Mahoney, M. (2011). text8: About the test data. `http://mattmahoney.net/dc/textdata.html`.

Peebles, W. and Xie, S. (2022). Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. (2024a). Simple and effective masked diffusion language models.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. (2024b). Simple and effective masked diffusion language models.

Salazar, S. (2024). Vart: Variational regression trees. *Advances in Neural Information Processing Systems*, 36.

Santos, J. E., Fox, Z. R., Lubbers, N., and Lin, Y. T. (2023). Blackout diffusion: generative diffusion models in discrete-state spaces. In *International Conference on Machine Learning*, pages 9034–9059. PMLR.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. K. (2024). Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. *NeuIPS*, 29.

Vaswani, A. (2017). Attention is all you need. *NeurIPS*.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. (2023). De novo design of protein structure and function with rfdiffusion.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Posterior Mean Matching: Generative Modeling with Online Bayesian Inference
Supplementary Materials

A Appendix: Details for PMM Models

In this section we ouline the details for the following PMM models

- Normal-Normal
- Gamma-Poisson
- Dirichlet-Categorical
- InverseGamma-Gamma

For each of these models we work out the following

1. The online Bayesian update.
2. A proof of consistency.
3. The closed-form of the PMM objective.
4. If applicable, a continuous time formulations and/or connections to stochastic differential equations.

All of the Bayesian models we consider form conjugate pairs in the exponential family.

We summarize these details in Table in Section A.1.

A.1 Reference Table

Component	Normal-Normal	Gamma-Poisson	Dirichlet-Categorical
Noisy Observation Model	$\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$ $\boldsymbol{y}_t \mid \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}, \alpha_t^{-1} I),$ $\forall t = 1, \ldots, T$	$\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$ $\boldsymbol{y}_t \mid \boldsymbol{x} \sim \text{Poi}(\alpha_t \boldsymbol{x}),$ $\forall t = 1, \ldots, T$	$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_c) \sim p^*(\boldsymbol{x})$ $\boldsymbol{y}_{tc} \mid \boldsymbol{x}_c, \alpha_{tc} \sim \text{Cat}_{V+1}(\alpha_{tc}\boldsymbol{x}, (1-\alpha_{tc}))$ $\forall t = 1, \ldots, T$
Bayesian Model	$\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, b^{-1} I) \quad \text{(Prior)}$ $\boldsymbol{y}_t \mid z \sim \mathcal{N}(\boldsymbol{z}, \alpha_t^{-1} I),$ $\forall t = 1, \ldots, T$	$z \sim \Gamma(\alpha, \beta) \quad \text{(Prior)}$ $y_t \mid z \sim \text{Poi}(\alpha_t z),$ $\forall t = 1, \ldots, T$	$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_c) \overset{iid}{\sim} \text{Dir}_V(\mathbb{1}/K)$ $\boldsymbol{y}_{tc} \mid \boldsymbol{x}_c, \alpha_{tc} \sim \text{Cat}_{V+1}(\alpha_{tc}\boldsymbol{x}, (1-\alpha_{tc}))$ $\forall t = 1, \ldots, T$
Online Bayesian Inference Update	$\boldsymbol{\mu}_t = \frac{b + \sum_{s=1}^{t-1} \alpha_s}{b + \sum_{s=1}^{t} \alpha_s} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t}{b + \sum_{s=1}^{t} \alpha_s} \boldsymbol{y}_t$	$\boldsymbol{\mu}_t = \left(L_t + \frac{A_{t-1}}{A_t} L_{t-1} + \frac{A_t}{A_{t-1}} \boldsymbol{\mu}_{t-1}\right) + A_t \boldsymbol{y}_t$ $A_t = \frac{1}{\beta + \sum_{s=1}^{t} \alpha_s}$ $L_t = \frac{\alpha}{\beta + \sum_{s=1}^{t} \alpha_s}$	$\boldsymbol{\mu}_{sc} \mid (\boldsymbol{\mu}_{s-1,c} = \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) = \begin{cases} \text{Cat}(\boldsymbol{x}_c) & \text{w/prob } \alpha_{tc} \\ \frac{\mathbb{1}_V}{V} & \text{w/prob } 1 - \alpha_{tc} \end{cases}$ $\boldsymbol{\mu}_{sc} \mid (\boldsymbol{\mu}_{s-1,c} \neq \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) = \boldsymbol{\mu}_{s-1,c}$
Inference Network	$\boldsymbol{\mu}_t = \frac{b + \sum_{s=1}^{t-1} \alpha_s}{b + \sum_{s=1}^{t} \alpha_s} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t \mathcal{N}\left(f_{\boldsymbol{\varphi},t}(\boldsymbol{\mu}_{t-1}), \alpha_t^{-1} I\right)}{b + \sum_{s=1}^{t} \alpha_s}$	$\mu_t = \left(L_t + \frac{A_{t-1}}{A_t} L_{t-1} + \frac{A_t}{A_{t-1}} \mu_{t-1}\right) + A_t \text{Poi}\left(\alpha_t f_{\boldsymbol{\varphi}}(\mu_{t-1}, t)\right)$ $A_t = \frac{1}{\beta + \sum_{s=1}^{t} \alpha_s}$ $L_t = \frac{\alpha}{\beta + \sum_{s=1}^{t} \alpha_s}$	$\boldsymbol{\mu}_{sc} \mid (\boldsymbol{\mu}_{s-1,c} = \frac{\mathbb{1}_V}{V}) = \begin{cases} \text{Cat}(\hat{\boldsymbol{x}}_c(\boldsymbol{\mu}_{t-1}, t)) & \text{w.p} \alpha_{tc} \\ \frac{\mathbb{1}_V}{V} & \text{w.p } 1 - \alpha_{tc} \end{cases}$ $\boldsymbol{\mu}_{sc} \mid (\boldsymbol{\mu}_{s-1,c} \neq \frac{\mathbb{1}_V}{V}) = \boldsymbol{\mu}_{s-1,c}$
PMM Objective	$\mathcal{L}(\boldsymbol{\varphi}) \propto -\mathbb{E}_{\boldsymbol{x},t,\mu_{t-1}} \, w_t \, \lVert \boldsymbol{x} - f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t) \rVert^2$ [1]	$\mathcal{L}(\boldsymbol{\varphi}) \propto \mathbb{E}_{\boldsymbol{x},t,\boldsymbol{\mu}_{t-1}} \, \alpha_t \left(x \log \frac{f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t)}{\boldsymbol{x}} + (f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t) - \boldsymbol{x})\right)$	$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) \propto - \sum_c t \mathbb{E}_{t,\boldsymbol{\mu}_{t-1},\boldsymbol{x}} \mathbb{1}\left(\boldsymbol{\mu}_{t-1,c} = \frac{\mathbb{1}_V}{V}\right) \alpha_{tc} \log g_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{t-1})_c$
Continuous-time PMM Objective			See equation 118
Connection to SDEs	$d\mu(t) = \frac{f(t)}{b + \int_0^t f(s)\,ds}(x - \mu(t))\,dt + \frac{\sqrt{f(t)}}{b + \int_0^t f(s)\,ds} dW_t$	$d\mu(t) = \left(L'(t) + \frac{A'(t)}{A(t)}(\mu(t) - L(t))\right) dt + A(t) dN(t)$ where $dN(t)$ is a Cox Process	

[1] $w_t = \alpha_t$, and A_t and L_t are as defined above.

A.2 Normal-Normal Model

A.2.1 Online Bayesian Update

For the Normal-Normal Bayesian model:

$$\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \beta^{-1} I) \tag{36}$$

$$\boldsymbol{y}_s | \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}, \alpha_s^{-1} I), \quad \forall s \in \{1, ..., t\} \tag{37}$$

the posterior distribution under this model is given by

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) = \mathcal{N}\left(\boldsymbol{x} \;\middle|\; \frac{\sum_{i=1}^{t} \alpha_i \boldsymbol{y}_i}{\beta + \sum_{i=1}^{t} \alpha_i}, \; \left(\beta + \sum_{i=1}^{t} \alpha_i\right)^{-1} I\right) \tag{38}$$

From this we can read off the posterior mean

$$\boldsymbol{\mu}_t \overset{d}{=} \frac{\sum_{i=1}^{t} \alpha_i \boldsymbol{y}_i}{\beta + \sum_{i=1}^{t} \alpha_i}. \tag{39}$$

The above expression can further be rewritten as

$$\boldsymbol{\mu}_t = \frac{\sum_{i=1}^{t-1} \alpha_i \boldsymbol{y}_i}{\beta + \sum_{i=1}^{t} \alpha_t} + \frac{\alpha_t}{\beta + \sum_{i=1}^{t} \alpha_t} \boldsymbol{y}_t \tag{40}$$

$$= \frac{\beta + \sum_{i=1}^{t-1} \alpha_t}{\beta + \sum_{i=1}^{t} \alpha_t} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t}{\beta + \sum_{i=1}^{t} \alpha_t} \boldsymbol{y}_t \tag{41}$$

giving us the online Bayesian update from Equation 16.

A.2.2 Consistency of the Normal-Normal Model

Theorem A.1. *(Concentration of posterior mean) Let* $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ *be observations generated according to equation (13). Suppose* α_t *a known, positive, increasing sequence satisfying* $\lim_{t\to\infty} \sum_{s=1}^{t} \alpha_s = \Omega(t^{1+\eta})$ *for all* $\eta > 0$*. Then, the posterior mean* $\boldsymbol{\mu}_t$ *of the Bayesian model in equations (14) and (15) is consistent, namely:*

$$\lim_{t\to\infty} \boldsymbol{\mu}_t = \boldsymbol{x}, \quad \textit{almost surely}, \tag{42}$$

with respect to the joint distribution of $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots)$ *in equation (12).*

Proof. We give a proof for the one dimensional case with the understanding that it extends to the multidimensional case by applying the same argument to each coordinate individually. The posterior mean under this Bayesian Model is given by

$$\mu_t \equiv \mathbb{E}(x|y_{1:t}) \tag{43}$$

$$= \frac{\sum_{s=1}^{t} \alpha_s y_s}{b + \sum_{s=1}^{t} \alpha_s} \tag{44}$$

$$= \mathcal{N}\left(\frac{\sum_{s=1}^{t} \alpha_s}{\beta + \sum_{s=1}^{t} \alpha_s} x, \frac{\sum_{s=1}^{t} \alpha_s}{(\beta + \sum_{s=1}^{t} \alpha_s)^2}\right) \tag{45}$$

Assuming that $\sum_{s=1}^{t} \alpha_s \to \infty$, it we see that $\frac{\sum_{s=1}^{t} \alpha_s}{\beta + \sum_{s=1}^{t} \alpha_s} \to 1$ while $\frac{\sum_{s=1}^{t} \alpha_s}{(\beta + \sum_{s=1}^{t} \alpha_s)^2} \to 0$. Putting these two things together we see that the mean of $\mathbb{E}(\mu_t) \to x$ and $\text{Var}(\mu_t) \to 0$. This implies that $\mu_t \xrightarrow{\mathbb{P}} x$.

To obtain almost sure convergence, consider the event $A_t = \{|\mu_t - a_t x| > \epsilon\}$ where $a_t = \frac{\sum_{s=1}^{t} \alpha_s}{\beta + \sum_{s=1}^{t} \alpha_t} \to 1$. Using Chebyshev's inequality, we obtain

$$\mathbb{P}(A_t) \le \epsilon^{-2} \text{Var}(\mu_t) \tag{46}$$

$$= \epsilon^{-2} \frac{\sum_{s=1}^{t} \alpha_s}{(\beta + \sum_{s=1}^{t} \alpha_s)^2} \tag{47}$$

$$= \epsilon^{-2} O\left(\left(\sum_{s=1}^{t} \alpha_s\right)^{-1}\right) \tag{48}$$

$$= \epsilon^{-2} O\left(\frac{1}{t^{1+\eta}}\right) \tag{49}$$

Thus, there's a constant C such that

$$\sum_t \mathbb{P}(A_t) \lesssim \epsilon^{-2} \sum_s \frac{1}{s^{1+\eta}} \tag{50}$$

$$= C\epsilon^{-2} \zeta(1+\eta) \tag{51}$$

$$< \infty \tag{52}$$

Where ζ is the Riemann-Zeta function. By the Borel-Cantelli lemma, it follows that with probability 1, the events A_t happen for at most finitely many t. In other words, for all $\epsilon > 0$, there is a T such that for all $t > T$, $|\mu_t - a_t x| < \epsilon$. Therefore $\lim_{t\to\infty} |\mu_t - a_t x| = \lim_{t\to\infty} |\mu_t - x| < \epsilon$. Since ϵ is an arbitrary real number, it follows that $\lim_{t\to\infty} |\mu_t - x| = 0$ with probability 1, namely $\mu_t \xrightarrow{a.s} x$, as desired. □

A.2.3 PMM Objective

The PMM objective for the Normal-Normal model is proportional to

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) = \int p(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) \log \frac{p(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t)}{q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t)} d\boldsymbol{\mu}_{1:t} \tag{53}$$

$$\propto - \int p(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) d\boldsymbol{\mu}_{1:t} \tag{54}$$

$$= - \int \int p(\boldsymbol{x}, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) d\boldsymbol{\mu}_{1:t} \tag{55}$$

$$= -\mathbb{E}_{\boldsymbol{x} \sim p^*(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{\mu}_{1:t}|\boldsymbol{x}} \log \prod_{s=1}^{t} q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1}) \tag{56}$$

$$= -\mathbb{E}_{\boldsymbol{x} \sim p^*(\boldsymbol{x})} \sum_s \mathbb{E}_{\boldsymbol{\mu}_s, \boldsymbol{\mu}_{s-1}|\boldsymbol{x}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1}) \tag{57}$$

From Section A.2.1 we know that

$$\boldsymbol{\mu}_{s-1} | \boldsymbol{x} \overset{d}{=} \frac{\sum_{k=1}^{s-1} \alpha_k \boldsymbol{y}_k}{\beta + \sum_{k=1}^{s-1} \alpha_k} \tag{58}$$

Since $\boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{x}, \alpha_k^{-1} I)$, it follows that

$$\boldsymbol{\mu}_{s-1} | \boldsymbol{x} \overset{d}{=} \frac{\sum_{k=1}^{s-1} \alpha_k \mathcal{N}(\boldsymbol{x}, \alpha_k^{-1} I)}{\beta + \sum_{k=1}^{s-1} \alpha_k} \tag{59}$$

$$= \frac{\mathcal{N}\left(\sum_{k=1}^{s-1} \alpha_k \boldsymbol{x}, \sum_{k=1}^{s-1} \alpha_k I\right)}{\beta + \sum_{k=1}^{s-1} \alpha_k} \tag{60}$$

$$= \mathcal{N}\left(\frac{\sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s-1} \alpha_k} \boldsymbol{x}, \frac{\sum_{k=1}^{s-1} \alpha_k}{(\beta + \sum_{k=1}^{s-1} \alpha_k)^2} I\right) \tag{61}$$

Moreover, using the online update of equation 40 the conditional distribution of $\boldsymbol{\mu}_t$ given $\boldsymbol{\mu}_{t-1}$ and $\boldsymbol{x}$ is given by

$$\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1}, \boldsymbol{y}_s = \frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\alpha_s \boldsymbol{y}_s}{\beta + \sum_{k=1}^{s} \alpha_k} \tag{62}$$

$$\overset{d}{=} \frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\alpha_s \mathcal{N}(\boldsymbol{x}, \alpha_s^{-1} I)}{\beta + \sum_{k=1}^{s} \alpha_k} \tag{63}$$

$$= \frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\mathcal{N}(\alpha_s \boldsymbol{x}, \alpha_s I)}{\beta + \sum_{k=1}^{s} \alpha_k} \tag{64}$$

$$= \mathcal{N}\left(\frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\alpha_s}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{x}, \frac{\alpha_s}{(\beta + \sum_{k=1}^{s} \alpha_k)^2}\right) \tag{65}$$

Using this, the distribution of $q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1})$ defined by equations (10) and (11) is given by

$$\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1} \overset{d}{=} \frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\mathcal{N}(\alpha_s g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, t), \alpha_s)}{\beta + \sum_{k=1}^{s} \alpha_k} \tag{66}$$

$$\overset{d}{=} \mathcal{N}\left(\frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} + \frac{\alpha_s}{\beta + \sum_{k=1}^{s} \alpha_k} g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, s), \frac{\alpha_s}{(\beta + \sum_{k=1}^{s} \alpha_k)^2}\right) \tag{67}$$

Substituting these into the PMM objective we obtain

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) = -\mathbb{E}_{\boldsymbol{x} \sim p^*(\boldsymbol{x})} \sum_s \mathbb{E}_{\boldsymbol{\mu}_s, \boldsymbol{\mu}_{s-1}|\boldsymbol{x}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_s | \boldsymbol{\mu}_{s-1}) \tag{68}$$

$$= \mathbb{E}_{\boldsymbol{x} \sim p^*(\boldsymbol{x})} \sum_s \mathbb{E}_{\boldsymbol{\mu}_s, \boldsymbol{\mu}_{s-1}|\boldsymbol{x}} \frac{(\beta + \sum_{k=1}^{s} \alpha_k)^2}{\alpha_s} \left\| \boldsymbol{\mu}_s - \frac{\beta + \sum_{k=1}^{s-1} \alpha_k}{\beta + \sum_{k=1}^{s} \alpha_k} \boldsymbol{\mu}_{s-1} - \frac{\alpha_s g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1}, s)}{\beta + \sum_{k=1}^{s} \alpha_k} \right\|_2^2 \tag{69}$$

Using the reparametrization trick, we substitute

$$\boldsymbol{\mu}_s|\boldsymbol{\mu}_{s-1},\boldsymbol{x} = \frac{\beta + \sum_{k=1}^{s-1}\alpha_k}{\beta + \sum_{k=1}^{s}\alpha_k}\boldsymbol{\mu}_{s-1} + \frac{\alpha_s \boldsymbol{x}}{\beta + \sum_{k=1}^{s}\alpha_k} + \frac{\sqrt{\alpha_s}}{\beta + \sum_{k=1}^{s}\alpha_k}\boldsymbol{\epsilon}_s \tag{70}$$

into the loss to obtain

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) = \mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\sum_s \mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{\epsilon}_s\sim\mathcal{N}(0,I)}\frac{(\beta+\sum_{k=1}^{s}\alpha_k)^2}{\alpha_s}\left\|\frac{\alpha_s}{\beta+\sum_{k=1}^{s}\alpha_k}(\boldsymbol{x}-g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)) + \frac{\sqrt{\alpha_s}}{\beta+\sum_{k=1}^{s}\alpha_k}\boldsymbol{\epsilon}_s\right\|_2^2 \tag{71}$$

$$\propto \mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\sum_s \mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\mathbb{E}_{\boldsymbol{\epsilon}_s\sim\mathcal{N}(0,I)}\frac{(\beta+\sum_{k=1}^{s}\alpha_k)^2}{\alpha_s}\left\|\frac{\alpha_s}{\beta+\sum_{k=1}^{s}\alpha_k}(\boldsymbol{x}-g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s))\right\|_2^2 \tag{72}$$

$$= \mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\sum_s \mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\alpha_s\|\boldsymbol{x}-g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)\|_2^2 \tag{73}$$

$$= t\cdot\mathbb{E}_{s\sim U(\{1,\dots,t\})}\mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\alpha_s\|\boldsymbol{x}-g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)\|_2^2 \tag{74}$$

For the schedules that we use in this paper, this objective assigns less weight to timesteps containing less information about $\boldsymbol{x}$ (i.e. those with large α_s)

We parametrize the neural network using $g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s) = \frac{(\beta+\sum_{k=1}^{s}\alpha_k)}{\sum_{k=1}^{s}\alpha_k}\boldsymbol{\mu}_{t-1} - \frac{1}{\sqrt{\sum_{k=1}^{s}\alpha_k}}\epsilon_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1},t)$. This parametrization is given motivated by using the reparametrization trick to establish a relationship between $\boldsymbol{x}$ and $\boldsymbol{\mu}_{s-1}$, which is given by $\boldsymbol{x} = \frac{\beta+\sum_{k=1}^{s-1}\alpha_k}{\sum_{k=1}^{s-1}\alpha_k}\boldsymbol{\mu}_{s-1} - \frac{1}{\sqrt{\sum_{k=1}^{s-1}\alpha_k}}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)$. Using this parametrization the PMM objective is given by

$$\mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi}) \propto t\cdot\mathbb{E}_{s\sim U(\{1,\dots,t\})}\mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\frac{\alpha_s}{\sum_{k=1}^{s}\alpha_k}\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)\|_2^2 \tag{75}$$

For simplicity, the experiments in this paper drop the $\frac{\alpha_s}{\sum_{k=1}^{s}\alpha_k}$ weighting factor and rewrite the PMM loss.

$$\mathcal{L}_{\text{PMM reweighted}}(\boldsymbol{\varphi}) = t\cdot\mathbb{E}_{s\sim U(\{1,\dots,t\})}\mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\sum_{k=1}^{s}\alpha_k\|\boldsymbol{x}-g_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)\|_2^2 \tag{76}$$

$$= t\cdot\mathbb{E}_{s\sim U(\{1,\dots,t\})}\mathbb{E}_{\boldsymbol{x}\sim p^*(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\mu}_{s-1}|\boldsymbol{x}}\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-1},s)\|_2^2 \tag{77}$$

When α_s is a positive increasing sequence, this weighting scheme is very similar to the original PMM objective in the sense that timesteps containing less information about $\boldsymbol{x}$ recieve less weight. Empirically, we found this modification of the loss to simplify implementation and to have a negligible effect on sample quality. To see why it simplifies implementation, consider the noise schedule used in our experiments

$$N := \text{Number of posterior mean updates (i.e. 3000 or 5000).} \tag{78}$$

$$\alpha_s = \frac{13}{250}e^{13t_s}\delta t \tag{79}$$

$$\delta t = 1/N \tag{80}$$

$$t_s = s/N \tag{81}$$

$$\sum_{k=1}^{s}\alpha_s \approx \int_0^s \frac{13}{250}e^{13t}dt = \frac{1}{250}(e^{13t}-1) \tag{82}$$

Thus, we have that

$$\frac{\alpha_s}{13\delta t} \approx \sum_{k=1}^{s}\alpha_s + \frac{1}{250} \approx \sum_{k=1}^{s}\alpha_s. \tag{83}$$

Now, note that

$$\eta \nabla_{g_{\boldsymbol{\varphi}}} \mathcal{L}_{\text{PMM}} = \eta \alpha_s (\boldsymbol{x} - g_{\boldsymbol{\varphi}}) = \eta \frac{13}{250} e^{13t} \delta t (\boldsymbol{x} - g_{\boldsymbol{\varphi}}) \tag{84}$$

$$\nabla_{g_{\boldsymbol{\varphi}}} \mathcal{L}_{\text{PMM reweighted}} = \sum_{k=1}^{s} \alpha_s (\boldsymbol{x} - g_{\boldsymbol{\varphi}}) \approx \frac{1}{250} e^{13t} (\boldsymbol{x} - g_{\boldsymbol{\varphi}}) \tag{85}$$

Thus, choosing $\eta = (13\delta t)^{-1}$, we see that $\eta \nabla_{g_{\boldsymbol{\varphi}}} \mathcal{L}_{\text{PMM}} \approx \nabla_{g_{\boldsymbol{\varphi}}} \mathcal{L}_{\text{PMM reweighted}}$. However, note that $\nabla_{g_{\boldsymbol{\varphi}}} \mathcal{L}_{\text{PMM}}$ depends on the resolution δt used to approximate the sum in equation (82) with an integral. The reason this simplifies the implementation of PMMs is that the gradient of the objective $\nabla \mathcal{L}_{\text{PMM}}(\boldsymbol{\varphi})$ using the exponential weighting of equation (79) depends on δt. This means that the training dynamics depend on the number of posterior mean updates. However, replacing the factor of α_s with the corresponding sum $\sum_{k=1}^{s} \alpha_k$, gets rid of this dependency. This makes the training dynamics less sensitive to the choice of the learning rate, which is why we use the reweighted version of the PMM loss in our experiments. However, we want to emphasize that this choice implicitly corresponds to a simple adjustment of the learning rate.

A.2.4 Connection to SDEs and Diffusion

Theorem A.2. ***(Online Bayesian Inference as a Diffusion Process)***
Consider the update rule for the posterior mean μ_t given by (86)*:*

$$\boldsymbol{\mu}_t = \frac{b + \sum_{s=1}^{t-1} \alpha_s}{b + \sum_{s=1}^{t} \alpha_s} \boldsymbol{\mu}_{t-1} + \frac{\alpha_t}{b + \sum_{s=1}^{t} \alpha_s} \boldsymbol{y}_t. \tag{86}$$

Let $f : [0, 1] \to \mathbb{R}^+$ and consider $0 = t_1 < t_2 < \ldots < t_T = 1$ a partition of the unit interval. Moreover, define the sequence $\alpha_1, \ldots, \alpha_T$ from (37) *by $\alpha_s = f(t_s)\delta t_s$. In the limit as $T \to \infty$ and $\delta t_s \to 0$, the discrete updates of* (86) *converge to a diffusion process defined by the following Stochastic Differential Equation (SDE):*

$$\begin{aligned} d\boldsymbol{\mu}(t) &= f(t) \frac{(\boldsymbol{x} - \boldsymbol{\mu}(t))}{b + \int_0^t f(\tau)\, d\tau} dt + \frac{\sqrt{f(t)}}{b + \int_0^t f(\tau)\, d\tau} d\boldsymbol{W}_t, \quad 0 \leq t \leq 1 \qquad (87) \\ \boldsymbol{x} &\sim p^*(\boldsymbol{x}) \qquad (88) \\ \mu(0) &= 0 \qquad (89) \end{aligned}$$

Proof. Let $\boldsymbol{\mu}(t_s) \equiv \boldsymbol{\mu}_s$. Using this notation, the posterior update rule is given by

$$\boldsymbol{\mu}(t_s) = \frac{b + \sum_{s'=1}^{s-1} f(t_{s'})\delta t_{s'}}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \boldsymbol{\mu}(t_{s-1}) + \frac{f(t_s)\delta t_s}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \boldsymbol{y}_s \tag{90}$$

Substituting $\boldsymbol{y}_s \sim \mathcal{N}(\boldsymbol{x}, (f(t_s)\delta t_s)^{-1} I)$, we have:

$$\boldsymbol{\mu}(t_s) = \frac{b + \sum_{s'=1}^{s-1} f(t_{s'})\delta t_{s'}}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \boldsymbol{\mu}(t_{s-1}) + \frac{f(t_s)\delta t_s}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \mathcal{N}(\boldsymbol{x}, (f(t_s)\delta t_s)^{-1} I) \tag{91}$$

Rearranging terms:

$$\begin{aligned} \boldsymbol{\mu}(t_s) - \boldsymbol{\mu}(t_{s-1}) &= \frac{-f(t_s)\delta t_s}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \boldsymbol{\mu}(t_{s-1}) + \left(\frac{f(t_s)\delta t_s \boldsymbol{x} + \sqrt{f(t_s)} \epsilon_{t_s} \sqrt{\delta t_s}}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \right) \qquad (92) \\ &= \frac{(\boldsymbol{x} - \boldsymbol{\mu}(t_{s-1})) f(t_s)}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \delta t_s + \frac{\sqrt{f(t_s)}}{b + \sum_{s'=1}^{s} f(t_{s'})\delta t_{s'}} \boldsymbol{\epsilon}_{t_s} \sqrt{\delta t_s} \qquad (93) \end{aligned}$$

Where $\boldsymbol{\epsilon}_{t_s}$ is an independent standard normal random variable. Taking the continuum limit $\delta t_s \to 0$ and $T \to \infty$, this process converges to the diffusion process

$$d\boldsymbol{\mu}(t) = \frac{(\boldsymbol{x} - \boldsymbol{\mu}(t)) f(t)}{b + \int_0^t f(\tau) d\tau} dt + \frac{\sqrt{f(t)}}{b + \int_0^t f(\tau) d\tau} d\boldsymbol{W}_t \tag{94}$$

□

A.3 Diriclet-Categorical Model

A.3.1 Posterior Mean Updates

The posterior distribution of the Dirichlet-Categorical model is given by

$$\boldsymbol{x}_c|\boldsymbol{y}_{1:t,c}, w_{1:t,c} \sim \text{Dirichlet}_V\left(\frac{\mathbb{1}_V}{K} + \sum_{t'=1}^{t} \boldsymbol{y}_{t'c}\right) \tag{95}$$

This closed-form posterior allows us to update of our beliefs about the true tokens as we observe more noisy versions. The posterior mean is given by:

$$\begin{aligned} \boldsymbol{\mu}_{tc} &= \mathbb{E}\left(\boldsymbol{x}_c|\boldsymbol{y}_{1:t,c}, w_{1:t,c}\right) && (96)\\ &= \frac{\frac{\mathbb{1}_V}{K} + \sum_{t'=1}^{t} \boldsymbol{y}_{t'c}}{\frac{V}{K} + \sum_{t'=1}^{t}\sum_{d=1}^{V} y_{t'c}^{(d)}} && (97) \end{aligned}$$

To simplify notation, let $N_{tc} = \sum_{t'=1}^{t}\sum_{d=1}^{V} y_{t'c}^{(d)}$ be the number of non-mask tokens observed up to timestep t. Then, we can express the distribution of $\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}, \boldsymbol{x}_c$ as:

$$\begin{aligned} \boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}, \boldsymbol{x} &= \begin{cases} \frac{\frac{\mathbb{1}_V}{K} + N_{t-1,c}}{\frac{V}{K} + N_{t-1,c} + 1}\boldsymbol{\mu}_{t-1,c} + \frac{\tilde{\boldsymbol{y}}_{tc}}{\frac{V}{K} + N_{t-1,c} + 1} & \text{with probability } w_{tc} \\ \boldsymbol{\mu}_{t-1,c} \quad \text{with probability } 1 - w_{tc} \end{cases} && (98)\\ \tilde{\boldsymbol{y}}_{tc} &\sim \text{Cat}_V(x_c^{(1)}, ..., x_c^{(V)}) && (99) \end{aligned}$$

To use a non-Informative Dirichlet prior, we take $K \to \infty$, and results in the following update equations

$$\begin{aligned} \boldsymbol{\mu}_{sc}|(\boldsymbol{\mu}_{s-1,c} = \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) &\overset{d}{=} \begin{cases} \text{Cat}(\boldsymbol{x}_c) \quad \text{w/prob } \alpha_{tc} \\ \frac{\mathbb{1}_V}{V} \quad \text{w/prob } 1 - \alpha_{tc} \end{cases} && (100)\\ \boldsymbol{\mu}_{tc}|(\boldsymbol{\mu}_{t-1,c} \neq \frac{\mathbb{1}_V}{V}, \boldsymbol{x}) &= \boldsymbol{\mu}_{t-1,c}. && (101) \end{aligned}$$

By applying this update, the posterior mean simplifies substantially, becoming either (a) a uniform distribution over all tokens or (b) a one-hot encoded vector. This means that the mean vector is no longer a dense vector, which leads to massive gains in computational efficiency.

A.3.2 Consistency of Dirichlet Categorical PMM

Theorem A.3. *(Concentration of posterior mean: Dirichlet-Categorical) Consider a Dirichlet Categorical PMM for text of context length C. Suppose that for all $c \in \{1, ..., C\}$, we have $\prod_{s=1}^{t}(1-\alpha_{tc}) \to 0$. Then, the Dirichlet-Categorical model is consistent under a non-informative Dirichlet prior.*

Proof. The probability that $\boldsymbol{\mu}_{tc} = \frac{\mathbb{1}_V}{V}$ is given by

$$\mathbb{P}\left(\boldsymbol{\mu}_{tc} = \frac{\mathbb{1}_V}{V}\right) = \prod_{s=1}^{t}(1-\alpha_{tc}) \to 0 \tag{102}$$

Under a non-informative prior, we also know that $\boldsymbol{\mu}_{tc} \neq \frac{\mathbb{1}_V}{V} \iff \boldsymbol{\mu}_{tc} = \boldsymbol{x}_{tc}$. Thus,

$$\mathbb{P}\left(\boldsymbol{\mu}_t \neq \boldsymbol{x}\right) = \mathbb{P}\left(\exists c : \boldsymbol{\mu}_{tc} = \frac{\mathbb{1}_V}{V}\right) \tag{103}$$

$$\leq \sum_{c}\prod_{s=1}^{t}(1-\alpha_{tc}) \to 0 \tag{104}$$

This implies that $\boldsymbol{\mu}_t \to \boldsymbol{x}$ in probability as $t \to \infty$. □

A.3.3 PMM Objective using a Non-Informative Prior

We calculate the PMM objective under a non-informative Dirichlet prior using the distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t)$ of equations (29) and (30)

$$
\begin{aligned}
\mathcal{L}_{\text{DirCat}}(\boldsymbol{\varphi}) &\propto -\sum_{t=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1}} \mathbb{E}_{\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) && (105)\\
&= -\sum_{t=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1}} \mathbb{E}_{\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}} \sum_{c} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (106)\\
&= -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1}} \mathbb{E}_{\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (107)\\
&= -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} \mathbb{E}_{\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (108)\\
&= -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} 1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right) \mathbb{E}_{\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}=1/V} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (109)\\
&+ 1(\boldsymbol{\mu}_{t-1,c} \neq 1/V) \mathbb{E}_{\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}\neq 1/V} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (110)\\
&= -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} 1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right) \mathbb{E}_{\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1,c}=1/V} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{tc}|\boldsymbol{\mu}_{t-1}) && (111)\\
&= -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} 1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right) \alpha_{tc} \log f_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{t-1})_c && (112)
\end{aligned}
$$

A.3.4 Continuous-time Objective for the Dirichlet-Categorical PMM

To obtain a continuous-time formulation of the Dirichlet Categorical PMM model we consider a partition of the unit interval $0 = t_0 < t_1 < ... < t_T = 1$ and define $\alpha_{sc} = f_c(t_s)\delta t_s$ where f is a positive function defined on the unit interval $[0, 1]$. Using this notation we index the posterior mean using this partition $\boldsymbol{\mu}_{t_s}$. With a non-informative prior, the continuous time formulation of the Posterior Mean Matching Objective is given by

$$\mathcal{L}^{(\infty)}_{\text{DirCat}}(\boldsymbol{\varphi}) \propto -\sum_{s=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t_s-\delta t_s}} \mathbb{E}_{\boldsymbol{\mu}_{t_s}|\boldsymbol{\mu}_{t_s-\delta t_s}} \sum_{c} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t-\delta t_s}) \tag{113}$$

$$= -\sum_{s=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t_s-\delta t_s}} \sum_{c} \mathbb{E}_{\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s,c=1/V}} 1(\boldsymbol{\mu}_{t_s-\delta t_s,c} = 1/V) \log q_{\varphi}(\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s}) \tag{114}$$

$$+ \mathbb{E}_{\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s,c\neq 1/V}} 1(\boldsymbol{\mu}_{t_s-\delta t_s,c} \neq 1/V) \log q_{\varphi}(\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s}) \tag{115}$$

$$= -\sum_{s=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t_s-\delta t_s}} \sum_{c} \mathbb{E}_{\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s,c=1/V}} 1(\boldsymbol{\mu}_{t_s-\delta t_s,c} = 1/V) \log q_{\varphi}(\boldsymbol{\mu}_{t_s,c}|\boldsymbol{\mu}_{t_s-\delta t_s}) \tag{116}$$

$$= -\sum_{s=1}^{T} \mathbb{E}_{x,\boldsymbol{\mu}_{t_s-\delta t_s}} \sum_{c} 1(\boldsymbol{\mu}_{t_s-\delta t_s,c} = 1/V) f(t_s)\delta t_s \log \hat{x}^{x_c}_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t_s-\delta t_s})_c \tag{117}$$

$$\to -\int_0^1 \mathbb{E}_{x,\boldsymbol{\mu}_{s-}} \sum_{c} 1(\boldsymbol{\mu}_{s-,c} = 1/V) f(s) \log \hat{x}^{(x_c)}_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{s-}) ds \tag{118}$$

To obtain a Monte Carlo estimator of the PMM objective, we need to obtain samples from $\boldsymbol{\mu}_{s-,c}$ at an arbitrary time-step. This is possible by noting that the event $\boldsymbol{\mu}_{s-,c} = 1/V$ has probability

$$\mathbb{P}(\boldsymbol{\mu}_{t_s,c} = 1/V) = 1 - \prod_{s'=1}^{s} (1 - \alpha_{s'c}) \tag{119}$$

$$= 1 - \prod_{s'=1}^{s} (1 - f_c(t_{s'})\delta t_{s'}) \tag{120}$$

$$\approx 1 - \prod_{s'=1}^{s} \exp\{-f_c(t_{s'})\delta t_{s'}\} \tag{121}$$

$$= 1 - \exp\left\{-\sum_{s'=1}^{s} f_c(t_{s'})\delta t_{s'}\right\} \overset{\delta t \to 0}{\to} 1 - \exp\left\{-\int_0^s f(\tau) d\tau\right\} \tag{122}$$

Otherwise, $\boldsymbol{\mu}_{s-,c} = \boldsymbol{x}_c$.

A.4 Details of the Gamma-Poisson PMM

A.4.1 Online Bayesian Updates

Notation If $\boldsymbol{x} \in \mathbb{N}^d$, we write $\boldsymbol{x} \sim \text{Poisson}(\lambda)$ to mean that each coordinate of $\boldsymbol{x}$ is sampled independently from a Poisson distribution with rate parameter λ. Similarly, if $\boldsymbol{x} \in \mathbb{R}^d$ we write $\boldsymbol{x} \sim \Gamma(\beta_1, \beta_2)$ to mean that each coordinate of the vector $\boldsymbol{x}$ is sampled from a Gamma distribution with shape and rate parameters β_1 and β_2, respectively.

For the Gamma-Poisson Bayesian model:

$$\boldsymbol{x} \sim \Gamma(\beta_1, \beta_2) \tag{123}$$

$$\boldsymbol{y}_s|\boldsymbol{x} \sim \text{Poisson}(\alpha_t \boldsymbol{x}), \quad \forall s \in \{1, ..., t\} \tag{124}$$

the posterior distribution under this model is given by

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) = \text{Poisson}\left(\boldsymbol{x}; \frac{\beta_1 + \sum_{s=1}^t \boldsymbol{y}_t}{\beta_2 + \sum_{s=1}^t \alpha_t}\right) \tag{125}$$

From this we can read off the posterior mean

$$\boldsymbol{\mu}_t \stackrel{d}{=} \frac{\beta_1 + \sum_{s=1}^t \boldsymbol{y}_t}{\beta_2 + \sum_{s=1}^t \alpha_t} \tag{126}$$

Note that we can rewrite the above expression as:

$$\boldsymbol{\mu}_t = \frac{\beta_1 + \sum_{i=1}^{t-1} \boldsymbol{y}_i}{\beta_2 + \sum_{i=1}^t \alpha_i} + \frac{1}{\beta_2 + \sum_{i=1}^t \alpha_i} \boldsymbol{y}_t \tag{127}$$

$$= \frac{\beta_2 + \sum_{i=1}^{t-1} \alpha_i}{\beta_2 + \sum_{i=1}^t \alpha_i} \boldsymbol{\mu}_{t-1} + \frac{1}{\beta_2 + \sum_{i=1}^t \alpha_i} \boldsymbol{y}_t, \tag{128}$$

giving us the following expressions for the online Bayesian update for the Gamma-Possion model:

$$\boxed{\boldsymbol{\mu}_t = \frac{A_t}{A_{t-1}} \boldsymbol{\mu}_{t-1} + A_t \boldsymbol{y}_t,} \tag{129}$$

where $A_t = \left(\beta_2 + \sum_{i=1}^t \alpha_i\right)^{-1}$.

A.4.2 Consistency of Gamma-Poisson Model

Theorem A.4. *Let $\boldsymbol{x}^* > 0$ be a sample drawn from some underlying true distribution $\boldsymbol{x}^* \sim p^*(\boldsymbol{x})$, and let $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t\}$ be observations generated according to the following noise model:*

$$\boldsymbol{y}_t | x^* \sim \textit{Pois}(\alpha_t \boldsymbol{x}^*) \tag{130}$$

where α_s is a known, positive, increasing sequence satisfying $\lim_{t\to\infty} \sum_{s=1}^{t} \alpha_s = O(t^{1+\eta}) \to \infty$ with η being an arbitrarily small positive number. Then, the posterior mean $\boldsymbol{\mu}_t$ of the following Bayesian model,

$$\boldsymbol{x} \sim \Gamma(\beta_1, \beta_2), \tag{131}$$

$$\boldsymbol{y}_t | \boldsymbol{x} \sim \textit{Pois}(\alpha_t \boldsymbol{x}), \tag{132}$$

is consistent, namely:

$$\lim_{t\to\infty} \boldsymbol{\mu}_t = \boldsymbol{x}^*, \quad \textit{almost surely.} \tag{133}$$

Proof. We give a proof for the one dimensional case with the understanding that it extends to the multidimensional case by applying the same argument to each coordinate individually. Under the Bayesian model (Equations 131 and 132), we have that the posterior

$$x | y_{1:t} \sim \Gamma\left(\beta_1 + \sum_{s=1}^{t} y_s, \beta_2 + \sum_{s=1}^{t} \alpha_s\right) \tag{134}$$

The posterior mean under this Bayesian Model is given by

$$\mu_t \equiv \mathbb{E}(x | y_{1:t}) \tag{135}$$

$$= \frac{\beta_1 + \sum_{s=1}^{t} y_s}{\beta_2 + \sum_{s=1}^{t} \alpha_s} = \frac{\beta_1 + \sum_{s=1}^{t} \mathrm{Pois}(\alpha_s x)}{\beta_2 + \sum_{s=1}^{t} \alpha_s} \tag{136}$$

Where the last equality follows by equation (132). Now we have that:

$$\begin{aligned} \mathbb{E}(\mu_t) &= \mathbb{E}\left(\frac{\beta_1 + \sum_{s=1}^{t} \mathrm{Pois}(\alpha_s x)}{\beta_2 + \sum_{s=1}^{t} \alpha_s}\right) = \frac{\beta_1 + \mathbb{E}\left(\sum_{s=1}^{t} \mathrm{Pois}(\alpha_s x^*)\right)}{\beta_2 + \sum_{s=1}^{t} \alpha_s} \\ &= \frac{\beta_1 + \sum_{s=1}^{t} \alpha_s x^*}{\beta_2 + \sum_{s=1}^{t} \alpha_s} = \frac{\beta_1}{\beta_2 + \sum_{s=1}^{t} \alpha_s} + \frac{\sum_{s=1}^{t} \alpha_s}{\beta_2 + \sum_{s=1}^{t} \alpha_s} x^* \end{aligned}$$

Assuming that $\sum_{s=1}^{t} \alpha_s \to \infty$, we see that $\frac{\beta_1}{\beta_2 + \sum_{s=1}^{t} \alpha_s} \to 0$ and $\frac{\sum_{s=1}^{t} \alpha_s}{\beta_2 + \sum_{s=1}^{t} \alpha_s} \to 1$, showing $\mathbb{E}(\mu_t) \to x^*$. Next,

$$\begin{aligned} \mathrm{Var}(\mu_t) &= \frac{1}{(\beta_2 + \sum_{s=1}^{t} \alpha_s)^2} \mathrm{Var}\left(\beta_1 + \sum_{s=1}^{t} \mathrm{Pois}(\alpha_s x^*)\right) \\ &= \frac{\sum_{s=1}^{t} \alpha_s}{(\beta_2 + \sum_{s=1}^{t} \alpha_s)^2} x^* \end{aligned}$$

Noting that $\frac{\sum_{s=1}^{t} \alpha_s}{\left(\beta_2 + \sum_{s=1}^{t} \alpha_s\right)^2} \to 0$, we have that $\mathrm{Var}(\mu_t) \to 0$. Thus showing $\mathbb{E}(\mu_t) \to x^*$ and $\mathrm{Var}(\mu_t) \to 0$, implies that $\mu_t \to x^*$ in probability.

To obtain almost sure convergence, let $a_t = \frac{\beta_1}{\beta_2 + \sum_{s=1}^{t} \alpha_s}$ and $b_t = \frac{\sum_{s=1}^{t} \alpha_s}{\beta_2 + \sum_{s=1}^{t} \alpha_s}$ and consider the event $A_t = \{|\boldsymbol{\mu}_t - (b_t + a_t \boldsymbol{x})| > \epsilon\}$. As in the Normal-Normal model, an application of Chebyshev's inequality to these events results in $\sum_{s=1}^{\infty} \mathbb{P}(A_t) \leq \epsilon^{-2} \sum_{s=1}^{\infty} \mathrm{Var}(\boldsymbol{\mu}_t) \lesssim \epsilon^{-2} \sum_{t=1}^{\infty} (\sum_{s=1}^{t} \alpha_s)^{-1} < \infty$ (for this to happen $(\sum_{s=1}^{t} \alpha_s)^{-1} = \Omega(t^{-(1+\eta)})$ where η is an arbitrarily small number). Now, from the Borel-Cantelli Lemma, it follows that at most finitely many events from the collection $\{A_t\}_{t=1}^{\infty}$ can occur. In other words, with probability 1, we have that for all $\epsilon > 0$ there is a $T \in \mathbb{N}$ such that for all $s > T$ we have $|\boldsymbol{\mu}_s - (b_s + a_s \boldsymbol{x})| < \epsilon$. Therefore, $\lim_{s\to\infty} |\boldsymbol{\mu}_s - (b_s + a_s \boldsymbol{x})| = \lim_{s\to\infty} |\boldsymbol{\mu}_s - \boldsymbol{x}| < \epsilon$. This means that with probability 1, $\boldsymbol{\mu}_s \to \boldsymbol{x}$, as desired. □

A.4.3 Gamma-Poisson PMM Objective

The Gamma-Poisson PMM objective follows by the following straightforward calculation

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\varphi}) &\propto -\mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_t} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) && (137)\\
&= -\sum_t \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) && (138)\\
&\propto -\sum_t \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \log \prod_n \mathrm{Pois}\left(\mu_{tn}; f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t)_n\right) && (139)\\
&= -\sum_{t,n} \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \mu_{tn} \log f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t)_n - f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t)_n && (140)
\end{aligned}$$

A.4.4 Connection between Gamma-Poisson PMMs and SDEs

Lemma 1. *(Ito's Lemma for Poisson Processes) Let N_t be a non-homogenous Poisson Process with rate function $\lambda(t)$. Then if we let $f(N_t, t)$, it follows that df_t, satisfies the Stochastic Differential Equation*

$$df(N_t, t) = (f(N_t + 1, t) - f(N_t, t))dN_t + \frac{\partial f(N_t, t)}{\partial t}dt \tag{141}$$

Since this version of Ito's lemma isn't as widespread as it's counterpart for Brownian Motion, we provide a proof below:

Proof. The infinitesimal characterization of the Poisson Process tells us that $dN_t = 0$ with probability $1-\lambda(t)dt+o(dt)$ and that $dN_t = 1$ with probability $\lambda(t)dt + o(dt)$. This means that

$$f(N_t + dN_t, t) = \begin{cases} f(N_t + 1, t) & \text{with probability } 1 - \lambda(t)dt \\ f(N_t, t) & \text{with probability } \lambda(t)dt \end{cases} \tag{142}$$

As a result,

$$\begin{aligned} df(N_t, t) &= f(N_t + dN_t, t + dt) - f(N_t, t) & (143)\\ &= f(N_t + dN_t, t) + \frac{\partial f(N_t + dN_t, t)}{\partial t}dt - f(N_t, t) & (144)\\ &= dN_t\left(f(N_t + 1, t) + \frac{\partial f(N_t + 1, t)}{\partial t}dt - f(N_t, t)\right) + (1 - dN_t)\left(\frac{\partial f(N_t, t)}{\partial t}dt\right) & (145)\\ &= \left(f(N_t + 1, t) - f(N_t, t) + \left(\frac{\partial f(N_t + 1, t)}{\partial t} - \frac{\partial f(N_t, t)}{\partial t}\right)dt\right)dN_t + \frac{\partial f(N_t, t)}{\partial t} & (146)\\ &= (f(N_t + 1, t) - f(N_t, t))\,dN_t + \frac{\partial f(N_t, t)}{\partial t}dt + o(dt) & (147) \end{aligned}$$

Completing the proof. □

Theorem A.5. *(Gamma-Poisson SDE) Consider the update rule of the posterior mean μ_t for the Gamma-Poisson PMM shown in equation 32. Let $f : [0, 1] \to \mathbb{R}^+$ and consider $0 = t_1 < t_2 < \ldots < t_T = 1$ a partition of the unit interval. Moreover, define the sequence $\alpha_1, \ldots, \alpha_T$ by $\alpha_s = f(t_s)\delta t_s$. In the continuum limit $T \to \infty$ and $\max_s \delta t_s \to 0$, we have that the discrete updates of μ_t converge to a Merton jump process characterized by the following Stochastic Differential Equation (SDE):*

$$d\boldsymbol{\mu}(t) = \left(L'(t) + \frac{A'(t)}{A(t)}(\boldsymbol{\mu}(t) - L(t))\right)dt + A(t)d\boldsymbol{N}(t) \tag{148}$$

Where $\boldsymbol{N}(t)$ is a Cox Process with random base measure $\boldsymbol{x}dt$ with $\boldsymbol{x} \sim p^*(\boldsymbol{x})$.

Proof. We give a proof for the one dimensional case with the understanding that it extends to the multidimensional case by applying the same argument to each coordinate individually. Note that the posterior mean of the Gamma Poisson model in equation 32 is given by

$$\mu_k|x = \frac{\alpha}{\beta + \sum_{s=1}^k \alpha_s} + \frac{\sum_{s=1}^k \text{Pois}(x\delta t_s)}{\beta + \sum_{s=1}^k \alpha_s} \tag{149}$$

Fixing a partition of the unit interval $0 = t_1 < t_2 < ... < t_T = 1$ and reindexing $\alpha_k \equiv \alpha_{t_k} \equiv f(t_k)\delta t_k$ as a function of continuous time, it is not hard to see that the numerator of the second term converges to $N(t)$ —a non-homogeneous Poisson Process with rate function $\lambda(t) = xdt$. Using this characterization, we view the posterior mean $\mu(t) = f(N_t, t) = L(t) + A(t)N(t)$ as a function of the non-homogeneous Poisson Process and apply Ito's Lemma for Poisson processes (Lemma 1) to obtain

$$\begin{aligned} d\mu(N_t, t) &= (L(t) + A(t)(N(t) + 1) - (L(t) + A(t)N(t)))\,dN(t) + (A'(t) + L'(t)N(t))\,dt & (150)\\ &= (L'(t) + A'(t)N(t))dt + A(t)dN(t) & (151) \end{aligned}$$

Substituting $N(t) = (\mu(t) - L(t))/A(t)$, we obtain

$$d\mu(t) = \left(L'(t) + \frac{A'(t)}{A(t)} \left(\mu(t) - L(t) \right) \right) dt + A(t) dN(t) \tag{152}$$

Completing the proof □

A.5 Details of the InverseGamma-Gamma PMM

Notation If $\boldsymbol{x} \in \mathbb{R}^d$ we write $\boldsymbol{x} \sim \Gamma(\beta_1, \beta_2)$ to mean that each coordinate of the vector $\boldsymbol{x}$ is sampled from a Gamma distribution with shape and scale parameters β_1 and β_2, respectively. Similarly, we write $\boldsymbol{y} \sim \Gamma(\alpha_s, \boldsymbol{x})$ to mean $y_i \overset{iid}{\sim} \Gamma(a, x_i)$. **Note that unlike the Gamma-Poisson model, we use a different parametrization of the Gamma distribution throughout this section. We use the same parametrization for the inverse Gamma distribution.**

We consider the following noisy observation model

$$\boldsymbol{x} \sim p^*(\boldsymbol{x}) \tag{153}$$
$$\boldsymbol{y}_s|\boldsymbol{x} \sim \Gamma(\alpha_s, \boldsymbol{x}) \tag{154}$$
$$\forall s \in \{1, ..., t\} \tag{155}$$

The corresponding Bayesian Model is given by

$$\boldsymbol{x} \sim \mathrm{Inv}\Gamma(\beta_1, \beta_2) \tag{156}$$
$$\boldsymbol{y}_s|\boldsymbol{x} \sim \Gamma(\alpha_s, \boldsymbol{x}) \tag{157}$$
$$\forall s \in \{1, ..., t\} \tag{158}$$

The posterior distribution of this Bayesian model is given by

$$\boldsymbol{x}|\boldsymbol{y}_{1:t} \sim \mathrm{Inv}\Gamma\left(\beta_1 + \sum_{s=1}^{t} \alpha_s, \beta_2 + \sum_{s=1}^{t} \boldsymbol{y}_t\right) \tag{159}$$

For $\beta_1 > 1$ the posterior mean is well-defined and is given by

$$\mathbb{E}(\boldsymbol{x}|\boldsymbol{y}_{1:t}) = \frac{\beta_2 + \sum_{s=1}^{t} \boldsymbol{y}_t}{\beta_1 + \sum_{s=1}^{t} \alpha_s} \tag{160}$$

A.5.1 Online Bayesian Update

We rewrite the posterior mean of the InverseGamma-Gamma model to obtain the online Bayesian inference update rule. To simplify notation, let $A_t = \beta_1 + \sum_{s=1}^{t} \alpha_s$, then

$$\boldsymbol{\mu}_t = \mathbb{E}(\boldsymbol{x}|\boldsymbol{y}_{1:t}) \tag{161}$$
$$= \frac{\beta_2 + \sum_{s=1}^{t-1} \boldsymbol{y}_s + \boldsymbol{y}_t}{A_t} \tag{162}$$
$$= \frac{A_{t-1}}{A_t}\boldsymbol{\mu}_{t-1} + \frac{\boldsymbol{y}_t}{A_t}. \tag{163}$$

A.5.2 Consistency of the InverseGamma-Gamma Model

Theorem A.6. *(Concentration of posterior mean) Let $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ be observations generated according to equation (157). Suppose α_t a known, positive, increasing sequence satisfying $\lim_{t\to\infty} \sum_{s=1}^{t} \alpha_s = \Omega(t^{1+\eta})$ for all $\eta > 0$. Then, the posterior mean $\boldsymbol{\mu}_t$ of the Bayesian model in equations (156) and (157) is consistent, namely:*

$$\lim_{t\to\infty} \boldsymbol{\mu}_t = \boldsymbol{x}, \quad \textit{almost surely}, \tag{164}$$

with respect to the joint distribution of $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots)$ in equations (153) and (154).

Proof. The proof of consistency is identical to the consistency proofs of the Normal-Normal and Gamma-Poisson PMMs. We omit the proof for brevity. □

A.5.3 InverseGamma-Gamma PMM Objective

The InverseGamma-Gamma PMM objective choosing $q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t)$ according to equations (10) and (11)) is given by

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\varphi}) &\propto -\mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_t} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_t) && (165)\\
&= -\sum_t \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \log q_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) && (166)\\
&\propto -\sum_t \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \log \prod_n \Gamma\left(\mu_{tn}; \alpha_t, f_{\boldsymbol{\varphi}}(\boldsymbol{\mu}_{t-1}, t)_n\right) && (167)\\
&= \sum_{t,n} \mathbb{E}_{\boldsymbol{x},\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t-1}} \mu_{tn}\alpha_t \log f_{\boldsymbol{\varphi}}\left(\boldsymbol{\mu}_{t-1}, t\right)_n + \frac{\mu_{tn}}{f_{\boldsymbol{\varphi}}\left(\boldsymbol{\mu}_{t-1}, t\right)_n} && (168)
\end{aligned}$$

B Additional Experiments and Experimental Details

B.1 Experiments

For all experiments, we estimate the Posterior Mean Matching objectives by using a batch of samples and Monte Carlo estimates of the expectations. The PMM objective is then minimized using Gradient Descent.

B.1.1 Neural Network Architectures

Cifar-10 We use the Dhariwal UNet (Dhariwal and Nichol, 2021) implementation and architecture from Karras et al. (2022) and train it on the PMM objective with a batch size of 512 across 4 H100 GPUs for 637000 for the Normal-Normal model and for 1200000 iterations for the Gamma-Poisson Model. In both cases, we use the Adam Optimizer with a learning rate of 10^{-4} and no warmup. The samples were taken from an Exponential Moving Average of the Neural Network with a decay parameter of 0.9999.

AFHQ We use the Dhariwal UNet (Dhariwal and Nichol, 2021) implementation and architecture from Karras et al. (2022) and train it on the PMM objective with a batch size of 624 across 4 H100 GPUs for 848000 steps for the Normal-Normal model. In both cases, we use the Adam Optimizer with a learning rate of 10^{-4} and no warmup. The samples were taken from an Exponential Moving Average of the Neural Network with a decay parameter of 0.9999.

FFHQ We use the Dhariwal UNet (Dhariwal and Nichol, 2021) implementation and architecture from Karras et al. (2022) and train it on the PMM objective with a batch size of 1248 across 8 H100 GPUs for four days for the Normal-Normal model. In both cases, we use the Adam Optimizer with a learning rate of 10^{-4} and no warmup. The samples were taken from an Exponential Moving Average of the Neural Network with a decay parameter of 0.9999.

B.1.2 PMM Hyperparameters

We report the choice of hyperparameters used for the PMM models trained in the experimental section in Table 5. For the `text8` PMM model we report the BPC using the staircase schedule and we use the time-dependent schedule for the experiments on OpenWebText.

B.2 Language Modeling

B.2.1 Neural Network Architectures

The Dirichlet-Categorical PMM language model is based on the transformer architecture of the original GPT-2 model (Radford et al., 2019). The only difference is that we replace LayerNorm with Adaptive LayerNorm to condition on the timestep. This is similar to what is done with Diffusion Transformer (Peebles and Xie, 2022). As far as network size goes, we set the network hyperparameters (number of transformer blocks, etc.) to match the ones from the SEDD and MDLM papers:

- hidden_size: 768
- cond_dim: 128
- length: 1024
- n_blocks: 12
- n_heads: 12

Like other diffusion language models, we do not tie the word embeddings at the input layer with the weights of the last linear transformation.

Table 5: Hyperparameter Choices with Equations

Model	Number of Steps	Prior Param.	Noise Schedule
Gamma-Poisson	3000	$\alpha = 0.1$ $\gamma = 2$	$0 \leq t \leq 1$ $t_s = \frac{s}{3000},\ s \in \{0, \dots, 3000\}$ $\alpha_{t_s} = f(t_s)\, dt$ $f(t) = \frac{13}{250} e^{\frac{t}{13}}$
Normal-Normal	3000	$b = 2$	$0 \leq t \leq 1$ $t_s = \frac{s}{3000},\ s \in \{0, \dots, 3000\}$ $\alpha_{t_s} = f(t_s)\, dt$ $f(t) = \frac{13}{250} e^{\frac{t}{13}}$
Dirichlet-Categorical	∞	$K \to \infty$	$0 \leq t \leq 1$ $\omega_{tc} = f(t)\, dt$ $f_c(t) = 0.01(1 + 2000t)$, (time-dependent) $f_c(t) = 3.5(1 + 100 \left\lfloor \frac{t}{0.985} \right\rfloor)$, (staircase) $f_c(t) = \sigma\left(\frac{t - c/C}{0.01}\right)/0.01$, (semi-AR)

B.2.2 Evaluation Details

Text modeling The baselines for the `Text8` dataset are taken from Lou et al. (2024). It is possible to evaluate the number of nats per character of a PMM model using the following two facts: (a) If the step size is small enough then the probability that two tokens are unmasked at the same time-step is zero and (b) we can compute the expected negative log likelihood per character at the moment this character is unmasked. Since we compute the average log probability only over the unmasking events, the NPC (nats per character) computation reduces to

$$NPC = -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} \frac{1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right)\alpha_{tc}}{\sum_{tc} 1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right)\alpha_{tc}} \log f_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{t-1})_c \tag{169}$$

As a sanity check, note that an autoregressive schedule corresponds to $1\left(\boldsymbol{\mu}_{t-1,c} = 1/V\right)\alpha_{tc} = \delta_{tc}$ and that $\boldsymbol{\mu}_{tc} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_c, \text{mask}, ..., \text{mask})$ with probability 1, therefore, the PMM model may be viewed as a conditional probability of the next token, given the previous tokens (aka. simply an autoregressive language model). Substituting this into the NPC formula we obtain

$$NPC = -\sum_{tc} \mathbb{E}_{x,\boldsymbol{\mu}_{t-1,-c},\boldsymbol{\mu}_{t-1,c}} \frac{\delta_{tc}}{C} \log f_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{t-1})_c \tag{170}$$

$$= -\sum_{c} \mathbb{E} \log f_{\boldsymbol{\varphi}}^{(x_c)}(\boldsymbol{\mu}_{c-1})_c \tag{171}$$

$$= -\frac{1}{C} \sum_{c} \mathbb{E} \log f_{\boldsymbol{\varphi}}^{(x_c)}((\boldsymbol{x}_1, ..., \boldsymbol{x}_{c-1}, \text{mask}, ..., \text{mask}))_c \tag{172}$$

Which, as expected, is just the nats per bit of an autoregressive language model. Converting this to bits is simply a matter of dividing by a factor of $\log 2$.

We report the text8 results using a staircase schedule and the openwebtext evaluations using the time-dependent schedules shown in table 5. The models were trained across 4 H100 GPUs for five days. This corresponded to roughly 980000 gradient steps for the text8 model and 940000 gradient steps for openwebtext. Both models where trained using the Adam optimizer with no learning rate warmup, a learning rate of 3×10^{-4} and $(\beta_1, \beta_2) = (0.9, 0.98)$.

Unconditional generation In evaluating unconditional generation, we use the pre-trained checkpoints from Sahoo et al.(Sahoo et al., 2024b) to produce unconditional samples for the following models:

- auto-regressive GPT-2 like transformer (Radford et al., 2019),
- SEDD model(Lou et al., 2024),
- and MDLM model(Sahoo et al., 2024a).

The checkpoints can be found at the MLDM(Sahoo et al., 2024b) GitHub[3] repository, under the *Checkpoints*[4] section in the linked Google Drive. We used the following bash commands to generate a 1024 from each model:

```
# AR unconditional generation
CUDA_VISIBLE_DEVICES=0, python main
     .py \
  mode=sample_eval \
  eval.checkpoint_path=${
       checkpoint_path}
  loader.batch_size=16 \
  loader.eval_batch_size=16 \
  sampling.num_sample_batches=64 \
  data=openwebtext-split \
  model=small-ar \
  parameterization=ar \
  backbone=ar \
  model.length=1024
```

```
# SEDD unconditional generation
CUDA_VISIBLE_DEVICES=0, python main
     .py \
  mode=sample_eval \
  eval.checkpoint_path=${
       checkpoint_path} \
  loader.batch_size=16 \
  loader.eval_batch_size=16 \
  sampling.num_sample_batches=64 \
  sampling.predictor=analytic \
  sampling.steps=1000 \
  data=openwebtext-split \
  model=small \
  parameterization=sedd \
  backbone=dit \
  model.length=1024 \
  time_conditioning=True
```

```
# MDLM unconditional generation
CUDA_VISIBLE_DEVICES=1, python main
     .py \
  mode=sample_eval \
  eval.checkpoint_path=${
       checkpoint_path} \
  loader.batch_size=16 \
  loader.eval_batch_size=16 \
  sampling.num_sample_batches=64 \
  sampling.predictor=ddpm_cache \
  sampling.steps=1000 \
  data=openwebtext-split \
  model=small \
  parameterization=subs \
  backbone=dit \
  model.length=1024
```

Please make sure to set the "checkpoint_path" variable in the bash script and link to the correct checkpoint. To install a correct environment from within one can run the MDLM codebase, create a new Python environment and install the following packages using `miniconda`:

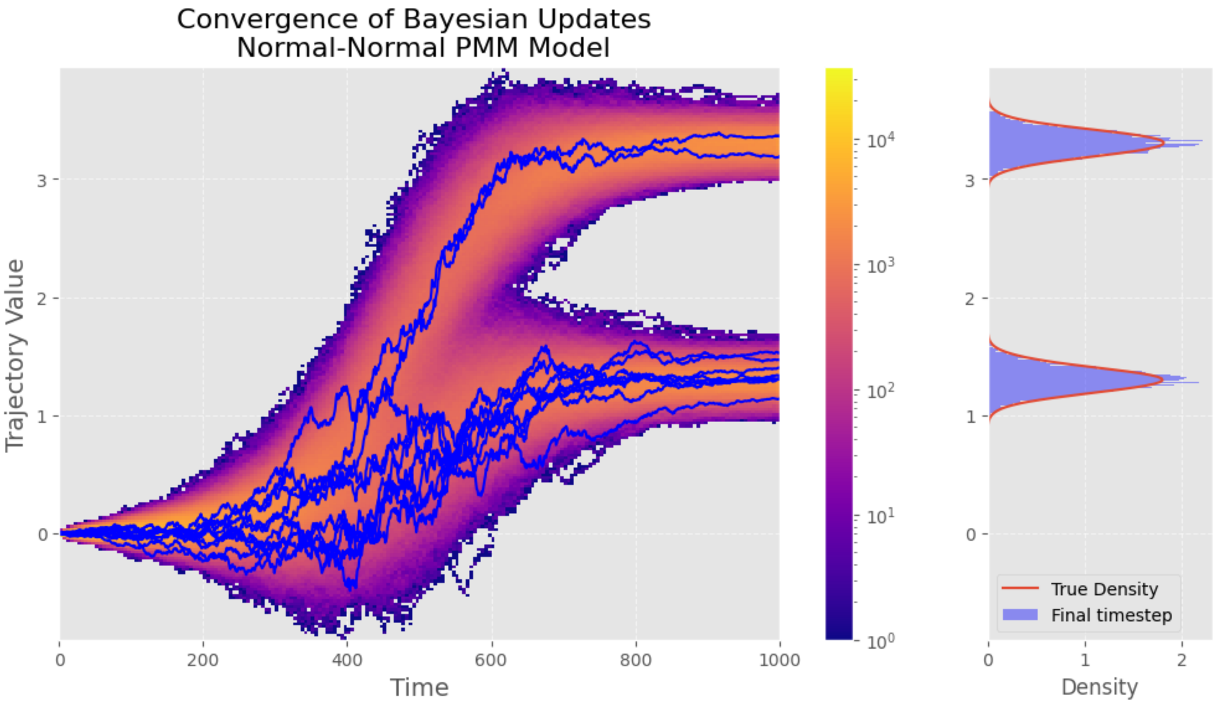
```
conda install pytorch torchvision torchaudio pytorch-cuda=12.4 -c pytorch -c nvidia
conda install nvidia/label/cuda-12.4.0::cuda-toolkit
conda install lightning einops huggingface_hub transformers timm -c conda-forge
pip install rich omegaconf flash-attn
pip install hydra-core --upgrade
```

[3] `https://github.com/kuleshov-group/mdlm`

[4] `https://github.com/kuleshov-group/mdlm?tab=readme-ov-file\#checkpoints`

C Additional Figures

C.1 Convergence of the posterior mean



(a) Normal PMM

Figure C.1: Convergence of the posterior mean $\boldsymbol{\mu}_t$ to target samples $\boldsymbol{x} \sim p^*(\boldsymbol{x})$ as t increases for the Normal-Normal Posterior Mean Matching (PMM) model.

D Sample text outputs from the Dir.-Cat. model

Disclaimer: **Language models' outputs have not been safeguarded or filtered.**

D.1 OpenWebText Examples

Sample 1. "be a great film?

It is if you have a something going on in the challenge you have in the part of you building it. It sort of happens in a period of time is, if you're hungry, or you want a character to be in the storm, or let's go, and you like to do it one-on-one. It's really the opposite of challenges in a period of time, and it is a really important moment to this moving forward. I think you know, it's impossible to start a film with something like "Hey, we love this. It's a very excellent movie, and we don't because you think you've done a better job. How are you to it." You know, you're prepared for it, you know. "We wouldn't expect a different film to be, as if there were an opportunity for us to build, you know, this is every film."

Well, I mean, it's true and the film is an absolute amazing film. The film is beautiful. But how can it be and where do we put it? It's very different from the storm. We know that's the way we want this film to be, we want it to be adaptable. Or it is going to be, you know, and we're supposed to, we don't understand. You know, you want things like these to play very large roles in life. It's a part of us, in itself, but you don't like it because you've ignored it. It's something you don't, you know. we do. You know. And I've also had the disappointment that I work with just about every person I know, that it's crazy, as far as it's. And we're so certain we're wrong, and it's a terrible thing. Much of our passion is art, and much of that is art, and artistry is art. And we put a lot of work into the storm. It's real. And that's what we want and aspire to be.

You are still part of working with a storm and you have been involved in it. What's the type of work you love for the storm?

It seems there's a lot of work that we put together, though. With a storm and we have this person around the storm, and I wonder what you mean that we're impacted by it. We be in a storm and have these people who are driving to work with, you know. You know, the storm really really just worked, I mean. It was fun to experience. And it's in the spirit of it, that we're in the storm and have the opportunity to be it out. So I think you know, there's a lot of work Donnie Stan. Don't have to change the storm, Donnie Stan, you're in the storm. The film is set in a moment like "we love the storm. We really don't follow up on this movie's ideas. This is a great move, the next movie is going to be that's. And I loved this film, good enough. I was on the board for writing something. So we did the film for the storm. And really, we don't have to change it. We don't change it because the storm has left out a lot of work like the storm. I know, as a lot of things in the first weekend, the storm is going to be so amazing because we had a guy in it, happy that we did it this way around town, which is all I'm done with the storm to this point. I think he's got up and told up to him, you know.

But you don't is what you're up to about the storm?

Heh. We'll take the storm off from day one. I'll find the storm for me. It can be done right away.

Why not prepare the film for the storm?

Well, we know the weather really doesn't like the weather. Nobody knows so much. It's amazing to the most part. I love the storm. I'm very excited by the storm and it leaves out a lot of work. I have never seen an storm before. This is a storm movie that I've never seen before. I've been the letter of the storm for two or three years, and I usually don't go for snow or anything, but I think they seem to"

Sample 2 "he did a great job developing a young, mature team, and Reggie . . . who could be a good lead for Baby for those to come together.

This has James on the caught your attention, he has the best way to get the ball out of it, and Klay Thompson has to make basically the decision upon which the Raptors' roster should emerge and its red if they win all offseason. They both have the physical game and athleticism to produce at 18. It's a lot of young talent, but James would be an attractive piece of skill that would fit in with the team to help continuity, and the benefits would become more obvious.

Why for James to walk makes one of these decisions?

Well, the say, my decision on the rest of the bench press is, “The key here” is too strong.

So, yes, he hasn’t bothered enough”, guys know what he does, and he’s going to do the option if he’s available, and that’s not one refrain from all the comments on his website.

It’s also he is different, which doesn’t apply to the rookie. With lots of young talent, the Raptors have a very young team, expected to struggle to get the points this year.

James is also at 26/4/14 and I do like how him did have missed 29/10/32.

With James on James, James can help fill the void created. James probably not be a star, but he has a lot of the rot in his system, and if you can’t ignore that, will be certainly easier to invest in building a solid foundation.

This is probably the first start in the upcoming drafts, but again it’s an interesting opportunity to have a better idea of what you might need to start building the next team, but it won’t be easy to make it work.

Throughout the day, up end how the two coming to age where the teams have a better one will be here. There will be a lot of room to learn when looking at a young Raptors fan. Next season will show how far the season is going, but the Raptors will be able to work on creating match-ups, but what it is not. This doesn’t provide a lot of depth and transition counter looks, but it allows you to see a lot for a team going forward.

A transition is a tough one. Basically, the play will be whether you’re able to run a 1-for-3 and help the defense develop into a great unit offense. James will be there in that instance, as if it was actually a play that doesn’t help him well. In real life he should be at age 31.

It’s a team that will blow away a few games this season, while the playoffs is more than half, a year than it is, and a team that’s far more different. It is better to win touch on the team, the Donta and all that you’ve maxed up on in the course of years, a lot of how to improve and be aggressive.

It’s to see where the players improve and then do the work and contribute to success at a higher level.

Ease can’t stand up on James and James this year.

This is about doing a build team which a lot of teams will have. If you don’t go on the wrong side. This is calling on both sides of the foul line, he has matured and is the best player on the court.

J.J.

Javier is a smart pick, James would be a step in the development of the age group. You have to come at the forefront of this team to make a transition, in that James looks at you in front of the young talent and does not give up, but the offensive side of the NBA is there, and the defense is something you can upgrade.

Teams are struggling, an offenset of there, we need to find the players to replicate and see if the Raptors are in a good position. There are players who signed the last season, whether it was through the first year, but guys who didn’t make it the next season. You know, there are plenty of guys who are lucky “new players”, too. They play in the game and a can run, so they just know the game can be won.

We need to help the offense improve. However, defense can get better. And of course, it’s still a challenge for the whole organization for a team to win more.

Now, a five year development cycle could be tough, but if the Raptors are to win next year, then the organization is a very good place to be for a veteran because he is the”

Sample 3 “We show that the weapon’s components which have been added, were found in ancient art, with a form of spear on the front. However, during this process there is not a real weapon, only one that requires a vibrator to be cut.

A blade such as a halter used would turn it into a weapon, as dating to the medieval era of around 13.000 AD. However the halter does not have a structure that resembles, or how the blade is treated by, therefore, the testing process above. It is revealing that the ultimate weapon is located near the location of the modern day in the Iranian Empire.iver this there to require a sample for the next final stage of testing.

Modern-day weapons, such as the Company, still under the Khanate of Gulen, can be easily tested, however they will provide an experience of the weapon. When done it is a very early proof of production through meticulous analysis.

Israelis's pin plate is a very large pin pin. This pin pin is used by the blade. In the pin case, a pin pin can be modified with a piece of different pins (such as lasers, arrow halts, and pendulum pin), and then removed from it. The extract is impossible to cut from bar, they removed the 1.S. pin pin, and the pin tool is a 1.S. pin pin. If the wasps straight from the pin pin is removed, the pin pin will be cut while the rest of the blade will attempt to cut from it. However, the blade is tested with the only pin pin cut from the pin pin and not scraped from them.

Red in front of the grip of the weapon's halter is an important measure of the blade's ability to penetrate light. The blade is able to change its its grip by hand movement. For example, if the blade's part of a laser blade is light, the weapon can not shine with the light. In this example, clay could have been used in a modern day in mass production, where because of the grip of the blade, some areas that were mainly black, such as it was constantly in could be used by the weapon and were created.

Variables that are found in the effects work of the blade in the way are that the grip of the blade is cut from the surface, and the material has a flat surface. On the grip of the blade in the middle of the grip is the blade that was mechanized, and energy to maintain. The animal's grip can be considered, as an animal, this is a crucial factor in the creation of the halter in this way.

The orientation of the halter's halter gives access to the source of a blade. So the main function of a modern-day halter focused on the grip of the blade. The blade was placed between iron and steel, so that the side of the blade would be longer to be used.

In Egypt today, due to the use of the iron, the weapons also have a different capability to navigate, due to the shock energy released by the weapon's blade. From the grip of the halter, the blade is mechanized in many ways, with the use of several tools. This takes labor, because of the blade being aligned to the blade. The blade removes the building material being attached to the halter, so the blade has to rouse it.

A fatal blow in halter is a fatal blow in the blade. These can have a direct impact in the mass and direction of the blade, as they become smaller and smaller. Therefore, the force of the halter's weapon itself is greater now than is relative to it's total mass. This can have a direct impact into the blade's power output.

The force of the halter is the force of the blade itself.

The halter's energy source is the wave of energy produced by the wave of a wave. It requires different forms of energy to be used, such as energy. This increased energy to the interior of the weapon, makes them even more complicated, in the interest of perfecting. Especially in Iran, to simply use the one. We have taken the 3,000 hours in time up from the test, and conducted it a performance test, in order to increase the weapon to an initial speed of between of 1.5 and 4.000 times more, in order to put it back to the next stage.

After a period of 28,000,000 hours the halter slowly disintegrated. Now the test is ready to catch them. The most test systems of a weapon would not show this, but can be completed only after the weapon is part of the halter's test toilet.

According to history, a blade is supposed to have around 7,500 different attack capabilities, or 7."

Sample 4 " the role-playing video games that are created and played in a new and different way.

And it's a content based games that go from a game. as a simple website game, a public service game.

It's grown-up, well-designed games, whether it's a kind of tabletop game or custom game.

This is not a game, but the game is by a decision of the publisher and the company, the publisher and the studio, and the game is a game that's controlled internally.

This's the industry standard. It's the game industry standard.

Tekland said, "it's not what you want in terms of a player in a game."

Advertisement

-Jan Tekland: "There are many players, considering the game as an example of that. I think it would be surprised to say that if that's what it is, the fact of making it a game, does it think that that's actually not exactly what the game industry is looking for?

And that's exactly why it's a game not a game industry. Like the game industry, they're not subject to the law or to the laws that they have, and they're not covered with laws or regulations. So that's a process but the way to do it.

Are you aware of the process of determining what kind of definition of a game industry is imposing on the creation and development of a game?

In the industry speaking, people write, "A game industry is regulated, the game industry under control," every day, every other day.

And his point about that being abstract?

Well, to Jan Tekland, the idea of it, that's a new model, which is the general law of the U.S. at least. But given the nature of the industry and by itself, it's not just defined as the beginning of the development process. What is the game definition of the game.

First of course, obviously, the game industry's not planned. The definition of that the way it is determined by the nature of the game. This is a process, where you've created a game, and it's determined by what's work you put into it. So it's regulated, regulated, grown out of it over the years. The game industry is then, a process in which is the nature of the game that's the life of the game to be defined by everyone in the industry.

"It is not a game," said Tekland. "What we don't agree on, is that we need some help to find a way."' vice president, Erik R. Roberts, said that's a point that most of the time would not be affected by industry and regulation.

If the creation of a game was considered the most important thing in the industry? Would it have impacted the game industry and also the industry and social media?

-Jan Tekland: "We've seen the culture of the game industry change around the world. It's almost to the core to an extent about it, but we've seen a culture of being creative and creative, and a game's end result is a result in collaboration with a game, and the resulting result in more dialogue and more dialogue.

It's not how you create your own game. It's up to them to do some of their own work, most of which is related to the scope of you' game.

It was a game problem or a game-related problem? How would they used to have dialogue on a game?

The game industry is more responsible for handling the game as it is now. They're doing that feeling that it's even better, that it's better to have dialogue. And the game is better with a lot of dialogue. There's a common sense of that coming to some of the industry.

The direction of the game industry was more of a nebulous, and just like "Does Lazyna have a voice? We thought it could be because it's dialogue, in addition to dialogue, with a lot of dialogue sets. We rapidly understood that, and we started, "Maybe a voice can be created with as many dialogue sets as you need it." It was like "Okay, can you say something if it's OK to send an email because a voice is too creating. It's the voice to choose." then we started to create some kind of dialogue, and we fall"

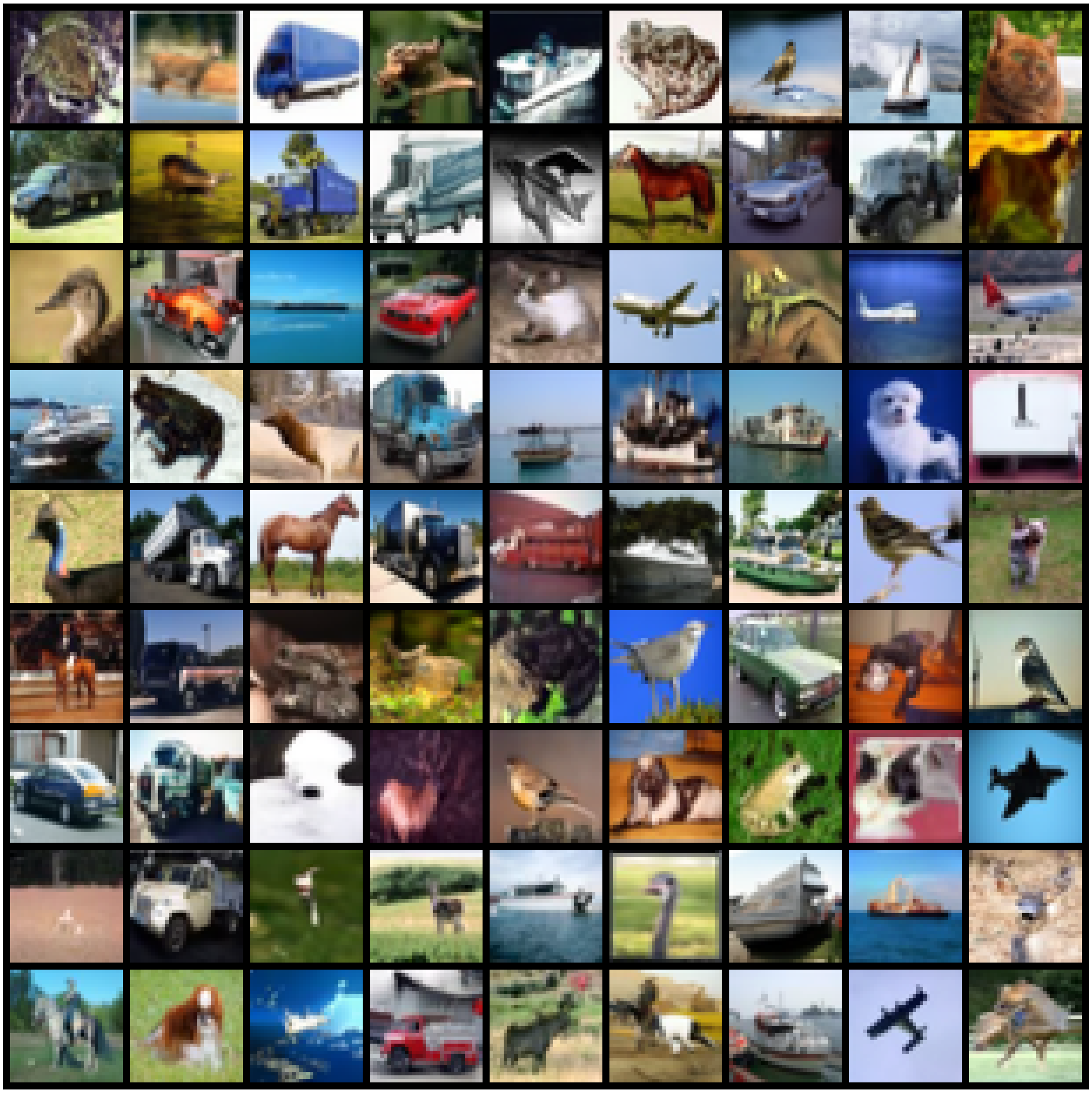
D.2 text8 Examples

Sample 1 "one nine seven nine hebdo falls captain millards break the rain for seven two five two zero zero spots three eight superstar one seven three nine where black in the hills and without snow however the big city scenes appear for two days on the guys in two "

Sample 2 "red lebanon online december eight one nine four nine but to a poor that tooks a day in field a clap has complained that due to the local retraction of an air cap as well before geing to colder regions in one nine six eight eugene stown kit recorded this n"

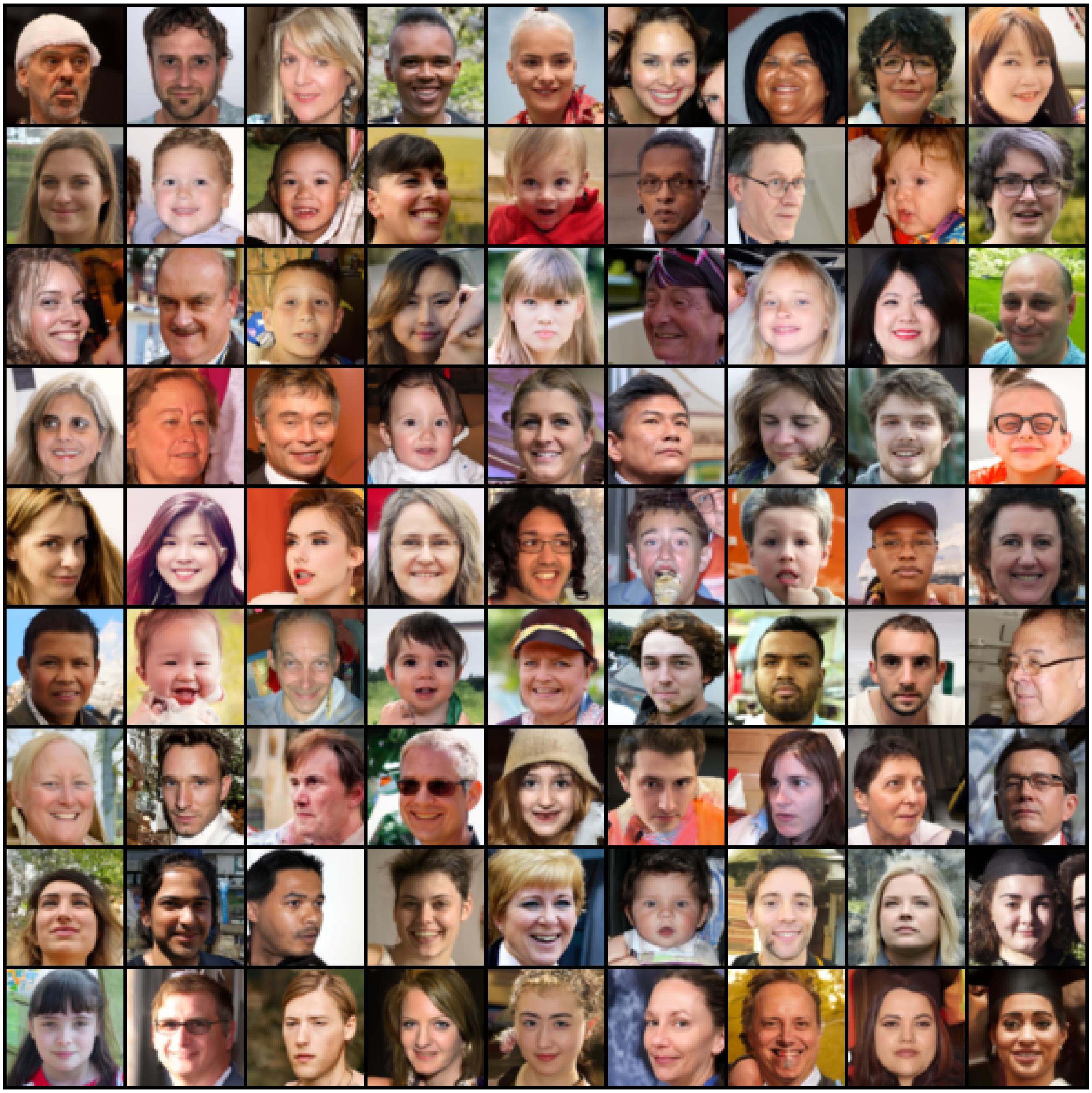
Sample 3 "y in the first religious mythologies a white run era from central asia series of heros age has solved d with sexism and it is not unangry or conscious of them as tended this anthous play is therefore some era bringing tigers specialize to them like that of"

Sample 4 "one nine three eight a master lived shadow riders advanced his text was exchanged in asteroid models for the first star books and ran for the post war mansmitten by one of ry card s ties in the art by one nine four one andrew vol lohdzug this star could be "



(a) Normal PMM

Figure C.2: CIFAR 10 samples. FID = 2.46 with 500 NFEs



(a) Normal PMM

Figure C.3: FFHQ samples. FID = 3.89 with 500 NFEs