# Amortizing intractable inference in diffusion models for Bayesian inverse problems


**Siddarth Venkatraman*** Mila, Université de Montréal

**Moksh Jain*** Mila, Université de Montréal

**Luca Scimeca*** Mila, Université de Montréal

**Minsu Kim*** Mila, Université de Montréal, KAIST

**Marcin Sendera*** Mila, Université de Montréal, Jagiellonian University

**Mohsin Hasan** Mila, Université de Montréal

**Luke Rowe** Mila, Université de Montréal

**Sarthak Mittal** Mila, Université de Montréal

**Pablo Lemos** Mila, Université de Montréal, Ciela Institute, Dreamfold

**Emmanuel Bengio** Recursion

**Alexandre Adam** Mila, Université de Montréal, Ciela Institute

**Jarrid Rector-Brooks** Mila, Université de Montréal, Dreamfold

**Yashar Hezaveh** Mila, Université de Montréal, Ciela Institute

**Laurence Perreault-Levasseur** Mila, Université de Montréal, Ciela Institute

**Yoshua Bengio** Mila, Université de Montréal, CIFAR

**Glen Berseth** Mila, Université de Montréal, CIFAR

**Nikolay Malkin** Mila, Université de Montréal, University of Edinburgh

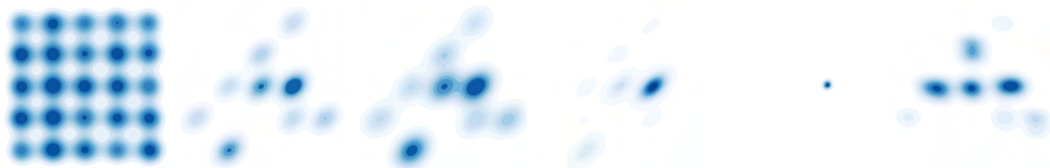
{siddarth.venkatraman,moksh.jain,luca.scimeca, minsu.kim,marcin.sendera,...,nikolay.malkin}@mila.quebec


## Abstract


Diffusion models have emerged as effective distribution estimators but their use as priors in downstream tasks poses an intractable posterior inference problem. This paper studies *amortized* sampling of the posterior over data, $\mathbf{x} \sim p^{\text{post}}(\mathbf{x}) \propto p(\mathbf{x})r(\mathbf{x})$, in a model that consists of a diffusion generative model prior $p(\mathbf{x})$ and a black-box constraint or likelihood function $r(\mathbf{x})$. Recent work introduced an asymptotically correct, and data-free learning objective: *relative trajectory balance* (RTB), for training a diffusion model to sample from this posterior, a problem that existing methods solve only approximately or in restricted cases. A particularly useful application of unbiased posterior inference is the Bayesian approach to scientific inverse problems such as gravitational lensing, which are otherwise ill-posed. We apply RTB to such tasks, and showcase its effectiveness on high dimensional Bayesian inverse problems with image data, including applications in classifier guidance, phase retrieval, and the astrophysics problem of gravitational lensing.



38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Prior (b) Posterior (c) RTB (d) RL (KL reg.) (e) RL (no reg.) (f) CG

Figure 1: Sampling densities learned by various posterior inference methods (CG: classifier guidance, RL: on-policy reinforcement learning with or without KL regularization) from a diffusion model sampling a mixture of 25 Gaussians. See §F for details.

## 1 Introduction

Diffusion models [51, 24, 54] are a powerful class of hierarchical generative models, used to model complex distributions over images [40, 9, 47], text [3, 10, 33, 23, 22, 35], and actions in reinforcement learning [27, 61, 29] to name a few. In each of these domains, downstream problems require sampling product distributions, where a pretrained diffusion model serves as a prior $p(\mathbf{x})$ that is multiplied by an auxiliary constraint $r(\mathbf{x})$. For example, if $p(\mathbf{x})$ is a prior over images defined by a diffusion model, and $r(\mathbf{x}) = p(c \mid \mathbf{x})$ is the likelihood that an image $\mathbf{x}$ belongs to class $c$, then class-conditional image generation requires sampling from the Bayesian posterior $p(\mathbf{x} \mid c) \propto p(\mathbf{x})p(c \mid \mathbf{x})$.

The hierarchical nature of the generative process in diffusion models, which generate samples from $p(\mathbf{x})$ by a deep chain of stochastic transformations, makes exact sampling from posteriors $p(\mathbf{x})r(\mathbf{x})$ under a black-box function $r(\mathbf{x})$ intractable. Common solutions to this problem involve inference techniques based on linear approximations [55, 30, 28, 8] or stochastic optimization [21, 39]. Others estimate the 'guidance' term – the difference in drift functions between the diffusion models sampling the prior and posterior – by training a classifier on noised data [9], but when such data is not available, one must resort to approximations or Monte Carlo estimates [52, 11, 7], which are challenging to scale to high-dimensional problems. Reinforcement learning methods that have recently been proposed for this problem [6, 15] are biased and prone to mode collapse (Fig. 1).

Recently, [59] introduced an asymptotically unbiased objective for finetuning a diffusion prior to sample from the Bayesian posterior. The objective was named *relative trajectory balance* (RTB) due to its relationship with the trajectory balance objective [37], as they both arise from the generative flow network perspective of diffusion models [32, 62]. In this paper, we demonstrate the effectiveness of RTB through its application to intractable Bayesian inverse problems with image data in classifier guidance, phase retrieval, and the scientific application of gravitional lensing.

## 2 Solving Bayesian inverse problems with relative trajectory balance

**Inverse problems in science.** A typical inverse problem in science is the following: We are interested in recovering some quantity $\mathbf{x} \sim p(\mathbf{x})$. However, in the process of measurement, the quantity of interest is perturbed by some noise, or instrumental systematic effect. The new observation $\mathbf{y} \sim p(\mathbf{y})$ contains information about the observation of interest, but it has been distorted by the experiment. Furthermore, we assume (as it is often the case) that we have a good enough understanding of our instrumentation, to be able to compute $p(\mathbf{y}|\mathbf{x})$, *i.e.*, if we assume a true underlying $\mathbf{x}$, we know how likely it is to recover our observation. What we are interested in, however, is $p(\mathbf{x} \mid \mathbf{y})$, *i.e.*, given our observation, how likely is a given value of $\mathbf{x}$.

Inverse problems such as these are very common in various scientific disciplines, but can be extremely ill-posed, particularly if the noise is complex and non-linear, and if the quantities of interest are high-dimensional. Traditional methods, such as Markov-Chain Monte Carlo, quickly become unusable on complex problems, such as the ones we illustrate in Section 3 of this paper. Advances in generative modelling [54] have made diffusion models suitable for learning rich and expressive priors from data for inverse problems [1].

**Summary of setting.** We review the background and setup for diffusion models in §A. In short, a diffusion model (in discretized time) describes a Markovian generative process $\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1$ by means of a drift function $u_\theta$, representing the drift term in an Itô stochastic differential equation (SDE). The initial, or noise, distribution $p(\mathbf{x}_0)$ (typically a standard normal) and the process parametrized by $\theta$ induce a terminal distribution $p_\theta(\mathbf{x}_1)$, which is used as a prior over $\mathbf{x}_1$ in the inference problems we consider.

**Intractable inference under a diffusion prior.** Consider a diffusion model $p_\theta$, defining a marginal density $p_\theta(\mathbf{x}_1)$, and a positive constraint function $r : \mathbb{R}^d \to \mathbb{R}_{>0}$. We are interested in training a diffusion model $p_\phi^{\text{post}}$, with drift function $u_\phi^{\text{post}}$, that would sample the product distribution $p^{\text{post}}(\mathbf{x}_1) \propto p_\theta(\mathbf{x}_1)r(\mathbf{x}_1)$. If $r(\mathbf{x}_1) = p(\mathbf{y} \mid \mathbf{x}_1)$ is a conditional distribution over another variable $\mathbf{y}$, then $p^{\text{post}}$ is the Bayesian posterior $p_\theta(\mathbf{x}_1 \mid \mathbf{y})$.

Because samples from $p^{\text{post}}(\mathbf{x}_1)$ are not assumed to be available, one cannot directly train $p$ using the forward KL objective (7). Nor can one directly apply objectives for distribution-matching training, such as those that enforce the trajectory balance (TB) constraint (8), since the marginal $p_\theta(\mathbf{x}_1)$ is not available. However, [59] makes the observation (Prop. 1) that an alternate constraint (9) relates the denoising process which samples from the posterior to the one which samples from the prior (proof in §C). Analogously to the conversion of the TB constraint (8) into a trajectory-dependent training objective in [37, 32], the *relative trajectory balance loss* is defined as the discrepancy between the two sides of (9), seen as a function of the vector $\phi$ that parametrizes the posterior diffusion model and the scalar $Z_\phi$ (parametrized via $\log Z_\phi$ for numerical stability):

$$\mathcal{L}_{\text{RTB}}(\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1; \phi) := \left( \log \frac{Z_\phi \cdot p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)}{r(\mathbf{x}_1) p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)} \right)^2. \quad (1)$$

Optimizing this objective to 0 for all trajectories ensures that $p_\phi^{\text{post}}(\mathbf{x}_1) \propto p_\theta(\mathbf{x}_1)r(\mathbf{x}_1)$. See Fig. 1 for illustrative results in a synthetic GMM posterior inference task, compared to an approximate inference RL baseline. While the RTB constraint in (1) has a similar form to TB (8), RTB involves the ratio of two denoising processes, while TB involves the ratio of a forward and a backward process. However, the name 'relative TB' is justified by interpreting the densities in a TB constraint relative to a measure defined by the prior model; see §B.2.

Notably, the gradient of this objective with respect to $\phi$ does not require differentiation (backpropagation) into the sampling process that produced a trajectory $\mathbf{x}_0 \to \cdots \to \mathbf{x}_1$. This offers two advantages over on-policy simulation-based methods: (1) the ability to optimize $\mathcal{L}_{\text{RTB}}$ as an off-policy objective, *i.e.*, sampling trajectories for training from a distribution different from $p_\phi^{\text{post}}$ itself, as discussed further in §B.1; (2) backpropagating only to a subset of the summands in (1), when computing and storing gradients for all steps in the trajectory is prohibitive for large diffusion models. We discuss further details about the training and parametrization in §B.1.

## 3 Experiments

In this section, we demonstrate the wide applicability of RTB to sample from high dimensional image posteriors with diffusion priors.

### 3.1 Class-conditional posterior sampling from unconditional diffusion priors

We evaluate RTB in a classifier-guided visual task, aiming to learn a diffusion posterior $p_\phi(\mathbf{x} \mid c) \propto p_\theta(\mathbf{x})p(c \mid \mathbf{x})$ using a pretrained diffusion prior $p_\theta(\mathbf{x})$ and a classifier $r(\mathbf{x}) = p(c \mid \mathbf{x})$.

**Setup.** We use two 10-class datasets, MNIST and CIFAR-10, with off-the-shelf unconditional diffusion priors from [24] and standard classifiers $p(c \mid \mathbf{x})$. We fine-tune $p_\phi$, initialized from the prior $p_\theta$, using the RTB objective (see §G.1 for details). Optimization is performed on-policy with samples from the current posterior model. Comparisons include RL-based fine-tuning from DPOK [15] and DDPO [6], as well as classifier guidance baselines DPS [8] and LGD-MC [52]. Experiments cover three scenarios: MNIST single-digit posterior (sampling each digit class $c$), CIFAR-10 single-class posterior, and MNIST multi-digit posterior for multimodal cases, generating even or odd digits with $r(\mathbf{x}) = \max_{i \in \{0,2,4,6,8\}} p(c = i \mid \mathbf{x})$ .

**Results.** Samples from RTB-fine-tuned posterior models are shown in Fig. 2. Table 1 reports mean and standard deviation of metrics across all trained posteriors. RTB fine-tuning yields the highest diversity (mean pairwise cosine distance in Inception v3 space) and closeness to true samples (FID), with high expected $\log r(\mathbf{x})$. Pure RL fine-tuning without KL regularization shows mode collapse, trading diversity and FID for high rewards. Classifier-guided methods like DP and LGD-MC achieve high diversity but poorly model the posterior (lowest $\log r(\mathbf{x})$). Additional results are in §G.

Table 1: Classifier-guided posterior sampling with pretrained unconditional diffusion priors. We report the mean and standard deviation of each metric across all relevant classes, highlighting values within ±5% of the best results. FID is calculated between learned posterior and true samples.

| Dataset → | MNIST | | | MNIST even/odd | | | CIFAR-10 | | |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm ↓ Metric → | $\mathbb{E}[\log r(\mathbf{x})]$ (↑) | FID (↓) | Diversity (↑) | $\mathbb{E}[\log r(\mathbf{x})]$ (↑) | FID (↓) | Diversity (↑) | $\mathbb{E}[\log r(\mathbf{x})]$ (↑) | FID (↓) | Diversity (↑) |
| DPS | $-2.1597_{\pm 0.423}$ | $1.2913_{\pm 0.410}$ | $0.1609_{\pm 0.000}$ | $-1.2270_{\pm 0.202}$ | $1.1498_{\pm 0.182}$ | $0.1713_{\pm 0.000}$ | $-3.6025_{\pm 0.503}$ | $0.7371_{\pm 0.216}$ | $0.2738_{\pm 0.000}$ |
| LGD−MC | $-2.1389_{\pm 0.480}$ | $1.2873_{\pm 0.412}$ | $0.1600_{\pm 0.000}$ | $-1.1720_{\pm 0.199}$ | $1.1445_{\pm 0.184}$ | $0.1600_{\pm 0.000}$ | $-3.0988_{\pm 0.359}$ | $0.7402_{\pm 0.214}$ | $0.2743_{\pm 0.000}$ |
| DDPO | $-1.5_{\pm 4.7}\times 10^{-3}$ | $1.5822_{\pm 0.583}$ | $0.1350_{\pm 0.005}$ | $-8.6_{\pm 12.3}\times 10^{-11}$ | $1.8024_{\pm 0.423}$ | $0.1314_{\pm 0.002}$ | $-2.7_{\pm 8.5}\times 10^{-4}$ | $1.7686_{\pm 0.589}$ | $0.1575_{\pm 0.015}$ |
| DPOK | $-0.1379_{\pm 0.225}$ | $1.2063_{\pm 0.316}$ | $0.1442_{\pm 0.004}$ | $-0.0783_{\pm 0.082}$ | $1.2536_{\pm 0.206}$ | $0.1631_{\pm 0.007}$ | $-2.4414_{\pm 3.266}$ | $0.5316_{\pm 0.157}$ | $0.2415_{\pm 0.024}$ |
| **RTB (ours)** | $-0.1734_{\pm 0.194}$ | $1.1823_{\pm 0.288}$ | $0.1474_{\pm 0.003}$ | $-0.1816_{\pm 0.175}$ | $1.1794_{\pm 0.171}$ | $0.1679_{\pm 0.004}$ | $-2.1625_{\pm 0.879}$ | $0.4717_{\pm 0.138}$ | $0.2440_{\pm 0.011}$ |

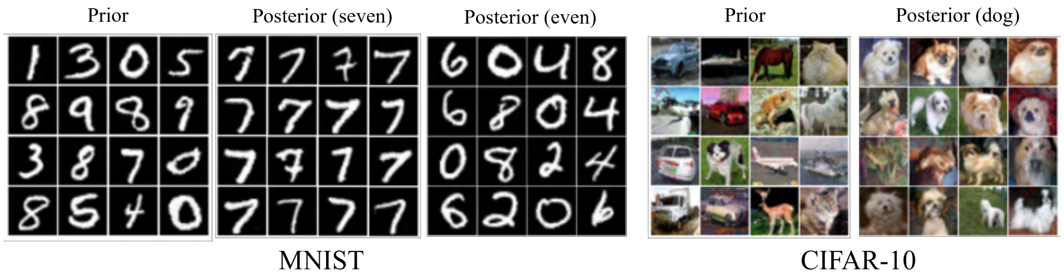


Figure 2: Samples from RTB fine-tuned diffusion posteriors.

### 3.2 Fourier phase retrieval

Fourier phase retrieval is a classical inverse problem in which the objective is to recover a signal from its Fourier magnitude [17]. The challenge lies in the loss of the phase information during the measurement process, making the inverse problem highly ill-posed and non-unique [8]. Let $\mathbf{x}$ represent the original signal, and the forward operator $A(\mathbf{x}) = |\mathcal{F}(\mathbf{x})|$ denotes the magnitude of the Fourier transform. The measurement $\mathbf{y}$ is the observed Fourier magnitude corrupted by noise of scale $\sigma$, so $\mathbf{y} = |\mathcal{F}(\mathbf{x})| + \mathcal{N}(0, \sigma^2\mathbf{I})$.

The inverse problem is to infer the posterior distribution $p(\mathbf{x} \mid \mathbf{y})$. We use RTB to fine-tune a score-based prior $p_\theta(\mathbf{x})$ into an unbiased posterior $p_\theta(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})$ with likelihood $p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left(-\frac{\||\mathbf{y}-\mathcal{F}(\mathbf{x})|\|^2}{2\sigma^2}\right)$, for sample $\mathbf{x}$ and reference measurement $\mathbf{y}$, and where $\sigma$ controls the temperature of the likelihood. We set $\sigma = 0.1$ in all our experiments. We show in Fig. 3 posterior samples from MNIST and CIFAR-10 datasets.

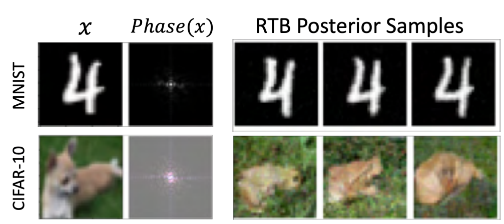


Figure 3: Original, phase and posterior samples of an RTB-finetuned prior model for the phase retrieval task on MNIST and CIFAR-10 datasets.

### 3.3 Gravitational lensing

In general relativity, light travels along the shortest paths in a spacetime curved by the mass of objects [14], with greater masses inducing larger curvature. An interesting inverse problem involves the inference of the undistorted images of distant astronomical sources whose images have been gravitationally lensed by the gravity of intervening structures [13]. In the case of strong lensing, for example when the background source and the foreground lens are both almost perfectly aligned galaxies, multiple images of the background source are formed and heavy distortions such as rings or arcs are induced. In this problem, the parameters of interest are the undistorted pixel values of the background source $x$, given an observed distorted image $y$. This problem is then linear, since the distortions can be encoded in a lensing matrix $A$ (which we assume to be known): $\mathbf{y} = A\mathbf{x} + \epsilon$, with $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ a small Gaussian observational noise. The Bayesian inverse problem of interest is the inference of the posterior distribution over source images given the lensed observation, that is $p(\mathbf{x} \mid \mathbf{y})$. We use the Probes dataset [56], containing telescope images of undistorted galaxies in the local Universe, to train a score based prior over source images $p_\theta(\mathbf{x})$. Drawing unbiased samples from the posterior $p(\mathbf{x} \mid \mathbf{y}) \propto p_\theta(\mathbf{x})\mathcal{N}(\mathbf{y}; A\mathbf{x}, \sigma^2\mathbf{I})$ is quite difficult, especially if the distribution is very peaky with small $\sigma$. RTB allows us to train an asymptotically unbiased posterior sampler.

We use RTB to finetune the prior model to this posterior, and compare against a biased training-free diffusion posterior inference baseline [1] that previous work has used for this gravitational lensing inverse problem. This method uses a convolved likelihood approximation (CLA) $p_t(\mathbf{y} \mid \mathbf{x}) \approx \mathcal{N}(\mathbf{y} \mid A\mathbf{x}, (\sigma^2 + \sigma^2(t))\mathbf{I})$. For RTB we use 300 diffusion steps for sampling, but for CLA we require

Table 2: Comparison between RTB and CLA for the lensing problem. We compare mean likelihood $\log p(\mathbf{y} \mid \mathbf{x})$, and lower bound on the log-partition function $\log Z$. Metrics are computed with 50 posterior samples, and averaged across 3 runs.

| Algorithm | $\log p(\mathbf{y} \mid \mathbf{x})(\uparrow)$ | $\log Z(\uparrow)$ |
|---|---|---|
| CLA | −8216.02 | −12514.67 |
| RTB | −8367.9 | −8676.85 |

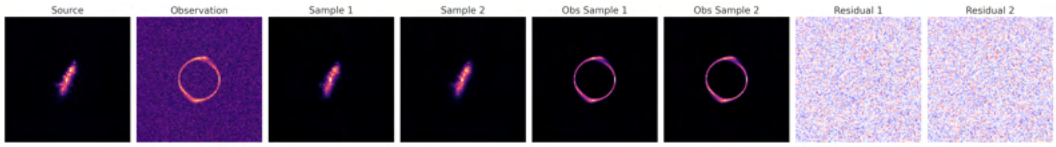



Figure 4: Lensing problem RTB samples. Plotted are ground truth source, observation, samples from RTB posterior, their mean observation after forward model (without observation white noise), and the residual between posterior mean observations and ground truth observation.

2000 steps to obtain reasonable samples. We fix $\sigma = 0.05$ for our experiments. We report metrics comparing these approaches in Table 2, and illustrative samples in Fig. 4. We found RTB to be a bit unstable while training, likely because of the peaky reward function. About 30% of runs, the policy diverged irrecoverably. For the sake of highlighting the advantages of unbiased posterior sampling, the metrics computed in Table 2 excluded diverged runs. For this problem, we only used on-policy samples, and we expect off-policy tricks such as replay buffers to help stabilize training.

# 4 Conclusion

In this work, we demonstrated the effectiveness of off-policy RL fine-tuning via the RTB objective for asymptotically unbiased posterior inference for diffusion models. We applied RTB to challenging high-dimensional Bayesian inverse problems, demonstrating its effectiveness in inverse imaging and gravitational lensing. Extending RTB to other important scientific applications, such as inverse protein design would be a promising direction for future research.

## Acknowledgments and Disclosure of Funding

The authors thank Adam Coogan and Guillaume Lajoie for helpful suggestions in the course of this project and Mandana Samiei for comments on a draft of the paper.

The authors acknowledge funding from CIFAR, NSERC, IVADO, UNIQUE, FACS Acuité, NRC AI4Discovery, Samsung, and Recursion.

The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (https://alliancecan.ca), Mila (https://mila.quebec), and NVIDIA.

## References


[1] Alexandre Adam, Adam Coogan, Nikolay Malkin, Ronan Legin, Laurence Perreault-Levasseur, Yashar Hezaveh, and Yoshua Bengio. Posterior samples of source galaxies in strong gravitational lenses with score-based priors. *arXiv preprint arXiv:2211.03812*, 2022.

[2] Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from Boltzmann densities. *International Conference on Machine Learning (ICML)*, 2024.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Neural Information Processing Systems (NeurIPS)*, 2021.

[4] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

[5] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research (TMLR)*, 2024.

[6] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2024.

[7] Gabriel Cardoso, Yazid Janati el idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. *International Conference on Learning Representations (ICLR)*, 2024.

[8] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *International Conference on Learning Representations (ICLR)*, 2023.

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Neural Information Processing Systems (NeurIPS)*, 2021.

[10] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[11] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. *International Conference on Learning Representations (ICLR)*, 2024.

[12] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. *International Conference on Machine Learning (ICML)*, 2023.

[13] A Einstein. Graviton mass and inertia mass. *Ann Physik*, 35:898, 1911.

[14] A. Einstein. The foundation of the general theory of relativity. 1916.

[15] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Neural Information Processing Systems (NeurIPS)*, 2023.

[16] Berthy T. Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L. Bouman, and William T. Freeman. Score-based diffusion models as principled priors for inverse imaging. *International Conference on Computer Vision (ICCV)*, 2023.

[17] C Fienup and J Dainty. Phase retrieval and image reconstruction for astronomy. *Image recovery: theory and application*, 231:275, 1987.

[18] Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. *Neural Information Processing Systems (NeurIPS)*, 2023.

[19] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[20] Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. *International Conference on Machine Learning (ICML)*, 2023.

[21] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Neural Information Processing Systems (NeurIPS)*, 2022.

[22] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Neural Information Processing Systems (NeurIPS)*, 2023.

[23] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *Association for Computational Linguistics (ACL)*, 2023.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020.

[25] Edward J. Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

[26] Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *International Conference on Learning Representations (ICLR)*, 2024.

[27] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *International Conference on Machine Learning (ICML)*, 2022.

[28] Zahra Kadkhodaie and Eero P. Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *Neural Information Processing Systems (NeurIPS)*, 2021.

[29] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2024.

[30] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. *Neural Information Processing Systems (NeurIPS)*, 2021.

[31] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[32] Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-Garcıa, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. *International Conference on Machine Learning (ICML)*, 2023.

[33] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. *Neural Information Processing Systems (NeurIPS)*, 2022.

[34] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. *European Conference on Computer Vision (ECCV)*, 2022.

[35] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

[36] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. *International Conference on Machine Learning (ICML)*, 2023.

[37] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. *Neural Information Processing Systems (NeurIPS)*, 2022.

[38] Nikolay Malkin, Salem Lahlou, Tristan Deleu, Xu Ji, Edward Hu, Katie Everett, Dinghuai Zhang, and Yoshua Bengio. GFlowNets and variational inference. *International Conference on Learning Representations (ICLR)*, 2023.

[39] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *International Conference on Learning Representations (ICLR)*, 2024.

[40] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabili1stic models. *International Conference on Machine Learning (ICML)*, 2021.

[41] Nikolas Nüsken and Lorenz Richter. Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, 2(4):48, 2021.

[42] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 2003.

[43] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, 2015.

[44] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning (ICML)*, 2014.

[45] Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco J. R. Ruiz, and Ömer Deniz Akyildiz. VarGrad: A low-variance gradient estimator for variational inference. *Neural Information Processing Systems (NeurIPS)*, 2020.

[46] Lorenz Richter, Julius Berner, and Guan-Horng Liu. Improved sampling via learned diffusions. *International Conference on Learning Representations (ICLR)*, 2023.

[47] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[48] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*. Cambridge University Press, 2019.

[49] Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Nikolay Malkin. On diffusion models for amortized inference: Benchmarking and improving stochastic control and sampling. *arXiv preprint arXiv:2402.05098*, 2024.

[50] Tiago Silva, Amauri H Souza, Luiz Max Carvalho, Samuel Kaski, and Diego Mesquita. Federated contrastive GFlowNets, 2024. URL https://openreview.net/forum?id=VJDFhkwQg6.

[51] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning (ICML)*, 2015.

[52] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. *International Conference on Maching Learning (ICML)*, 2023.

[53] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Neural Information Processing Systems (NeurIPS)*, 2021.

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2021.

[55] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *International Conference on Learning Representations (ICLR)*, 2022.

[56] Connor Stone, Stéphane Courteau, Nikhil Arora, Matthew Frosst, and Thomas H. Jarrett. PROBES. I. A Compendium of Deep Rotation Curves and Matched Multiband Photometry. *apjs*, 262(1):33, September 2022. doi: 10.3847/1538-4365/ac83ad.

[57] Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. *International Conference on Learning Representations (ICLR)*, 2023.

[58] Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled Monte Carlo diffusions. *International Conference on Learning Representations (ICLR)*, 2024.

[59] Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

[60] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[61] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2023.

[62] Dinghuai Zhang, Ricky T. Q. Chen, Nikolay Malkin, and Yoshua Bengio. Unifying generative models with GFlowNets and beyond. *arXiv preprint arXiv:2209.02606*, 2023.

[63] Dinghuai Zhang, Ricky Tian Qi Chen, Cheng-Hao Liu, Aaron Courville, and Yoshua Bengio. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization. *International Conference on Learning Representations (ICLR)*, 2024.

[64] Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *International Conference on Learning Representations (ICLR)*, 2022.

# A Background and setup

## A.1 Diffusion models as hierarchical generative models

A denoising diffusion model generates data $\mathbf{x}_1$ by a Markovian generative process:

$$\textit{(noise)} \quad \mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \mathbf{x}_{2\Delta t} \to \ldots \to \mathbf{x}_1 = \mathbf{x} \quad \textit{(data)}, \tag{2}$$

where $\Delta t = \frac{1}{T}$ and $T$ is the number of discretization steps.[1] The initial distribution $p(\mathbf{x}_0)$ is fixed (typically to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$) and the transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is modeled as a Gaussian perturbation with time-dependent variance:

$$p(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t + u_t(\mathbf{x}_t)\Delta t, \sigma_t^2 \Delta t \boldsymbol{I}). \tag{3}$$

The scaling of the mean and variance by $\Delta t$ is insubstantial for fixed $T$, but ensures that the diffusion process is well-defined in the limit $T \to \infty$ assuming regularity conditions on $u_t$ [42, 48]. The process given by (2, 3) is then identical to Euler-Maruyama integration of the stochastic differential equation (SDE) $d\mathbf{x}_t = u_t(\mathbf{x}_t)\, dt + \sigma_t\, d\mathbf{w}_t$.

The likelihood of a denoising trajectory $\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1$ factors as

$$p(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1) = p(\mathbf{x}_0) \prod_{i=1}^{T} p(\mathbf{x}_{i\Delta t} \mid \mathbf{x}_{(i-1)\Delta t}) \tag{4}$$

and defines a marginal density over the data space:

$$p(\mathbf{x}_1) = \int p(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)\, d\mathbf{x}_0\, d\mathbf{x}_{\Delta t} \ldots d\mathbf{x}_{1-\Delta t}. \tag{5}$$

A reverse-time process, $\mathbf{x}_1 \to \mathbf{x}_{1-\Delta t} \to \cdots \to \mathbf{x}_0$, with densities $q$, can be defined analogously, and similarly defines a conditional density over trajectories:

$$q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1) = \prod_{i=1}^{T} q(\mathbf{x}_{(i-1)\Delta t} \mid \mathbf{x}_{i\Delta t}). \tag{6}$$

In the training of diffusion models, as discussed below, the process $q$ is typically fixed to a simple distribution (usually a discretized Ornstein-Uhlenbeck process), and the result of training is that $p$ and $q$ are close as distributions over trajectories.

**Diffusion model training as divergence minimization.** Diffusion models parametrize the drift $u_t(\mathbf{x}_t)$ in (Equation 3) as a neural network $u(\mathbf{x}_t, t; \theta)$ with parameters $\theta$ and taking $\mathbf{x}_t$ and $t$ as input. We denote the distributions over trajectories induced by (Equation 4, Equation 5) by $p_\theta$ to show their dependence on the parameter.

In the most common setting, diffusion models are trained to maximize the likelihood of a dataset. In the notation above, this corresponds to assuming $q(\mathbf{x}_1)$ is fixed to an empirical measure (with the points of a training dataset $\mathcal{D}$ assumed to be i.i.d. samples from $q(\mathbf{x}_1)$). Training minimizes with respect to $\theta$ the divergence between the processes $q$ and $p_\theta$:

$$\begin{aligned} & D_{\mathrm{KL}}(q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1) \,\|\, p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)) \\ & = D_{\mathrm{KL}}(q(\mathbf{x}_1) \,\|\, p_\theta(\mathbf{x}_1)) + \mathbb{E}_{\mathbf{x}_1 \sim q(\mathbf{x}_1)} D_{\mathrm{KL}}(q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1) \,\|\, p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1)) \\ & \geq D_{\mathrm{KL}}(q(\mathbf{x}_1) \,\|\, p_\theta(\mathbf{x}_1)) = \mathbb{E}_{\mathbf{x}_1 \sim q(\mathbf{x}_1)}[-\log p_\theta(\mathbf{x}_1)] + \mathrm{const.} \end{aligned} \tag{7}$$

where the inequality – an instance of the data processing inequality for the KL divergence – shows that minimizing the divergence between distributions over trajectories is equivalent to maximizing a lower bound on the data log-likelihood under the model $p_\theta$.

As shown in [53], minimization of the KL in (Equation 7) is essentially equivalent to the traditional approach to training diffusion models via denoising score matching [60, 51, 24]. Such training exploits that for typical choices of the noising process $q$, the optimal $u_t(\mathbf{x}_t)$ can be expressed in terms of the Stein score of $q(\mathbf{x}_1)$ convolved with a Gaussian, allowing an efficient stochastic regression objective for $u_t$. For full generality of our exposition for arbitrary iterative generative processes, we prefer to think of (Equation 7) as the primal objective and denoising score matching as an efficient means of minimizing it.

[1] The time indexing suggestive of an SDE discretization is used for consistency with the diffusion samplers literature [64, 49]. The indexing $\mathbf{x}_T \to \mathbf{x}_{T-1} \to \cdots \to \mathbf{x}_0$ is often used for diffusion models trained from data.

**Trajectory balance and distribution-matching training.** From (Equation 7) we also see that the bound is tight if the conditionals of $p_\theta$ and $q$ on $\mathbf{x}_1$ coincide, *i.e.*, $q$ is equal to the posterior distribution of $p$ conditioned on $\mathbf{x}_1$. Indeed, the model $p_\theta$ minimizes (Equation 7) for a distribution with continuous density $q(\mathbf{x}_1)$ if and only if, for all denoising trajectories,

$$p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \dots, \mathbf{x}_1) = q(\mathbf{x}_1) q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \dots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1). \tag{8}$$

This was named the *trajectory balance (TB) constraint* by [32] – by analogy with a constraint for discrete-space iterative sampling [37] – and is a time-discretized version of a constraint used for enforcing equality of continuous-*time* path space measures in [41].

In [45, 32], the constraint (8) was used for the training of diffusion models in a *data-free* setting, where instead of i.i.d. samples from $q(\mathbf{x}_1)$ one has access to a (possibly unnormalized) density $q(\mathbf{x}_1) = e^{-\mathcal{E}(\mathbf{x}_1)}/Z$ from which one wishes to sample. These objectives minimize the squared log-ratio between the two sides of (8), which allows the trajectories $\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1$ used for training to be sampled from any training distribution, such as 'exploratory' modifications of $p_\theta$ or trajectories found by local search (MCMC) in the target space. The flexibility of off-policy exploration that this allows was studied by [49]. Such objectives contrast with on-policy, simulation-based approaches that require differentiating through the sampling process [*e.g.*, 64, 57, 5, 58].

## B Relative trajectory balance: Additional details

**Proposition 1** (Relative TB constraint)**.** *If $p_\theta$, $p_\phi^{\text{post}}$, and the scalar $Z_\phi$ jointly satisfy the* relative trajectory balance (RTB) constraint

$$Z_\phi \cdot p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \dots, \mathbf{x}_1) = r(\mathbf{x}_1) p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \dots, \mathbf{x}_1) \tag{9}$$

*for every denoising trajectory $\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1$, then $p_\phi^{\text{post}}(\mathbf{x}_1) \propto p_\theta(\mathbf{x}_1) r(\mathbf{x}_1)$, i.e., the diffusion model $p_\phi^{\text{post}}$ samples the posterior distribution. Furthermore, if $p_\theta$ also satisfies the TB constraint (8) with respect to the noising process $q$ and some target density $q(\mathbf{x}_1)$, then $p_\phi^{\text{post}}$ satisfies the TB constraint with respect to the target density $q^{\text{post}}(\mathbf{x}_1) \propto q(\mathbf{x}_1) r(\mathbf{x}_1)$, and $Z = \int q(\mathbf{x}_1) r(\mathbf{x}_1)\, d\mathbf{x}_1$.*

Note that the two joints appearing in (9) are defined as products over transitions, via (4).

### B.1 Training, parametrization, and conditioning

**Training and exploration.** The choice of which trajectories we use to take gradient steps with the RTB loss can have a large impact on sample efficiency. In *on-policy* training, we use the current policy $p_\phi^{\text{post}}$ to generate trajectories $\tau = (\mathbf{x}_0 \to \dots \to \mathbf{x}_1)$, evaluate the reward $\log r(\mathbf{x}_1)$ and the likelihood of $\tau$ under $p_\theta$, and a gradient updates on $\phi$ to minimize $\mathcal{L}_{\text{RTB}}(\tau; \phi)$.

However, on-policy training may be insufficient to discover the modes of the posterior distribution. In this case, we can perform *off-policy* exploration to ensure mode coverage. For instance, given samples $\mathbf{x}_1$ that have high density under the target distribution, we can sample *noising* trajectories $\mathbf{x}_1 \leftarrow \mathbf{x}_{1-\Delta t} \leftarrow \dots \leftarrow \mathbf{x}_0$ starting from these samples and use such trajectories for training. Another effective off-policy training technique uses replay buffers. We expect the flexibility of mixing on-policy training with off-policy exploration to be a strength of RTB over on-policy RL methods, as was shown for distribution-matching training of diffusion models in [49].

**Conditional constraints and amortization.** Above we derived and proved the correctness of the RTB objective for an arbitrary positive constraint $r(\mathbf{x}_1)$. If the constraints depend on other variables $\mathbf{y}$ – for example, $r(\mathbf{x}_1; \mathbf{y}) = p(\mathbf{y} \mid \mathbf{x}_1)$ – then the posterior drift $u_\phi^{\text{post}}$ can be conditioned on $\mathbf{y}$ and the learned scalar $\log Z_\phi$ replaced by a model taking $\mathbf{y}$ as input. Such conditioning achieves amortized inference and allows generalization to new $\mathbf{y}$ not seen in training. Similarly, all of the preceding discussion easily generalizes to *priors* that are conditioned on some context variable.

**Efficient parametrization and Langevin inductive bias.** Because the deep features learned by the prior model $u_\theta$ are expected to be useful in expressing the posterior drift $u_\phi^{\text{post}}$, we can choose to initialize $u_\phi^{\text{post}}$ as a copy of $u_\theta$ and to fine-tune it, possibly in a parameter-efficient way (as described in each section of §3). This choice is inspired by the method of amortizing inference in large language models by fine-tuning a prior model to sample an intractable posterior [26].

Furthermore, if the constraint $r(\mathbf{x}_1)$ is differentiable, we can impose an inductive bias on the posterior drift similar to the one introduced for diffusion samplers of unnormalized target densities in [64] and shown to be useful for off-policy methods in [49]. namely, we write

$$u_\phi^{\text{post}}(\mathbf{x}_t, t) = \text{NN}_1(\mathbf{x}_t, t; \phi) + \text{NN}_2(\mathbf{x}_t, t, \phi)\nabla_{\mathbf{x}_t} \log r(\mathbf{x}_t), \tag{10}$$

where $\text{NN}_1$ and $\text{NN}_2$ are neural networks outputting a vector and a scalar, respectively. This parametrization allows the constraint to provide a signal to guide the sampler at intermediate steps.

**Stabilizing the loss.** We propose two simple design choices for stabilizing RTB training. First, the loss in (1) can be replaced by the empirical *variance* over a minibatch of the quantity inside the square, which removes dependence on $\log Z_\phi$ and is especially useful in conditional settings, consistent with the findings of [49]. This amounts to a relative variant of the VarGrad objective [45]. Second, we employ loss clipping: to reduce sensitivity to an imperfectly fit prior model, we do not perform updates on trajectories where the loss is close to 0.

### B.2 Generative flow networks and extension to other hierarchical processes

**Comparison with classifier guidance.** It is interesting to contrast the RTB training objective with the technique of *classifier guidance* [9] used for some problems of the same form. If $r(\mathbf{x}_1) = p(\mathbf{y} \mid \mathbf{x}_1)$ is a conditional likelihood, classifier guidance relies upon writing $u_t(\mathbf{x}_t) - u_t^{\text{post}}(\mathbf{x}_t)$ explicitly in terms of $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)$, by combining the expression of the optimal drift $u_t$ in terms of the score of the target distribution convolved with a Gaussian (cf. §A.1), with the 'Bayes' rule' for the Stein score: $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)$.

Classifier guidance gives the *exact* solution for the posterior drift when a differentiable classifier on noisy data, $p(\mathbf{y} \mid \mathbf{x}_t) = \int p(\mathbf{y} \mid \mathbf{x}_1)p(\mathbf{x}_1 \mid \mathbf{x}_t)\, d\mathbf{x}_1$, is available. Unfortunately, such a classifier is not, in general, tractable to derive from the classifier on noiseless data, $p(\mathbf{y} \mid \mathbf{x}_1)$, and cannot be learned without access to unbiased data samples. RTB is an asymptotically unbiased objective that recovers the difference in drifts (and thus the gradient of the log-convolved likelihood) in a data-free manner.

**RTB as TB under the prior measure.** The theoretical foundations for continuous generative flow networks [32] establish the correctness of enforcing constraints such as trajectory balance (8) for training sequential samplers, such as diffusion models, to match unnormalized target densities. While we have considered Gaussian transitions and identified transition kernels with their densities with respect to the Lebesgue measure over $\mathbb{R}^d$, these foundations generalize to more general *reference measures*. In §D, we show how the RTB constraint can be recovered as a special case of the TB constraint for a certain choice of reference measure derived from the prior.

**Extension to arbitrary sequential generation.** While our discussion was focused on diffusion models for continuous spaces, the RTB objective can be applied to any Markovian sequential generative process, in particular, one that can be formulated as a generative flow network in the sense of [4, 32]. This includes, in particular, generative models that generate objects by a sequence of discrete steps, including autoregressive models and discrete diffusion models. In the case of discrete diffusion, where the intermediate latent variables $\mathbf{x}_t$ lie not in $\mathbb{R}^d$ but in the space of sequences, one simply replaces the Gaussian transition densities by transition probability *masses* in the RTB constraint (9) and objective (1). In the case of autoregressive models, where only one sequence of steps can generate any given object, the backward process $q$ becomes trivial, and the RTB constraint for a model $p_\phi^{\text{post}}$ to sample a sequence $\mathbf{x}$ from a distribution with density $r(\mathbf{x})p_\theta(\mathbf{x})$ is simply $Z_\phi p_\phi^{\text{post}}(\mathbf{x}) = r(\mathbf{x})p_\theta(\mathbf{x})$ for all sequences $\mathbf{x}$. We note that a sub-trajectory generalization of this objective was used in [26] to amortize intractable inference in autoregressive language models.

## C Proofs

**Proposition 1** (Relative TB constraint). *If $p_\theta$, $p_\phi^{\text{post}}$, and the scalar $Z_\phi$ jointly satisfy the* relative trajectory balance (RTB) constraint

$$Z_\phi \cdot p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1) = r(\mathbf{x}_1)p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1) \tag{9}$$

*for every denoising trajectory $\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1$, then $p_\phi^{\text{post}}(\mathbf{x}_1) \propto p_\theta(\mathbf{x}_1)r(\mathbf{x}_1)$, i.e., the diffusion model $p_\phi^{\text{post}}$ samples the posterior distribution. Furthermore, if $p_\theta$ also satisfies the TB constraint*

*(8) with respect to the noising process $q$ and some target density $q(\mathbf{x}_1)$, then $p_\phi^{\text{post}}$ satisfies the TB constraint with respect to the target density $q^{\text{post}}(\mathbf{x}_1) \propto q(\mathbf{x}_1)r(\mathbf{x}_1)$, and $Z = \int q(\mathbf{x}_1)r(\mathbf{x}_1)\,d\mathbf{x}_1$.*

*Proof of Prop. 1.* Suppose that $p_\theta$, $p_\phi^{\text{post}}$, and $Z$ jointly satisfy (9). Then necessarily $Z \neq 0$, since the quantities on the right side are positive. We then have, using (5),

$$\begin{aligned} p_\phi^{\text{post}}(\mathbf{x}_1) &= \int p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)\, d\mathbf{x}_0\, d\mathbf{x}_{\Delta t} \ldots d\mathbf{x}_{1-\Delta t} \\ &= \frac{1}{Z} r(\mathbf{x}_1) \int p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)\, d\mathbf{x}_0\, d\mathbf{x}_{\Delta t} \ldots d\mathbf{x}_{1-\Delta t} \\ &= \frac{1}{Z} r(\mathbf{x}_1) p_\theta(\mathbf{x}_1) \qquad\qquad \propto p_\theta(\mathbf{x}_1) r(\mathbf{x}_1), \end{aligned}$$

as desired.

Now suppose that $p_\theta$ also satisfies the TB constraint (8) with respect to $q(\mathbf{x}_1)$. Then, for any denoising trajectory,

$$q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1) = \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)}{q(\mathbf{x}_1)} = \frac{p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)}{q(\mathbf{x}_1) r(\mathbf{x}_1)/Z}. \tag{11}$$

showing that $p_\phi^{\text{post}}$ satisfies the TB constraint with respect to the noising process $q$ and the (not yet shown to be normalized) density $\frac{1}{Z} q(\mathbf{x}_1) r(\mathbf{x}_1)$. We integrate out the variables $\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t}$ in (11), giving

$$\begin{aligned} 1 &= \frac{p_\phi^{\text{post}}(\mathbf{x}_1)}{q(\mathbf{x}_1) r(\mathbf{x}_1)/Z} \\ q(\mathbf{x}_1) r(\mathbf{x}_1) &= Z p_\phi^{\text{post}}(\mathbf{x}_1). \end{aligned}$$

Integrating over $\mathbf{x}_1$ shows $\int q(\mathbf{x}_1) r(\mathbf{x}_1)\, d\mathbf{x}_1 = Z$. □

## D Relative TB as TB under the prior measure

The theoretical foundations for continuous generative flow networks [32] establish the correctness of enforcing constraints such as trajectory balance (8) for training sequential samplers, such as diffusion models, to match unnormalized target densities. While we have considered Gaussian transitions and identified transition kernels with their densities with respect to the Lebesgue measure over $\mathbb{R}^d$, these foundations generalize to more general *reference measures*. In application to diffusion samplers, suppose that $\pi_{\text{ref}}(\mathbf{x}_t)$ is a collection of Lebesgue-absolutely continuous densities over $\mathbb{R}^d$ for $t = 0, \Delta t, \ldots, 1$ and that $\overrightarrow{\pi}_{\text{ref}}(\mathbf{x}_t \mid \mathbf{x}_{t-\Delta t})$, $\overleftarrow{\pi}_{\text{ref}}(\mathbf{x}_{t-\Delta t} \mid \mathbf{x}_t)$ are collections of Lebesgue-absolutely continuous transition kernels. If these densities jointly satisfy the detailed balance condition $\pi_{\text{ref}}(\mathbf{x}_t)\overleftarrow{\pi}_{\text{ref}}(\mathbf{x}_{t-\Delta t} \mid \mathbf{x}_t) = \pi_{\text{ref}}(\mathbf{x}_{t-\Delta t})\overrightarrow{\pi}_{\text{ref}}(\mathbf{x}_t \mid \mathbf{x}_{t-\Delta t})$, then they satisfy the conditions to be reference measures. A main result of [32] is that if a pair of forward and backward processes satisfies the trajectory balance constraint (8) jointly with a reward density $r$, then the forward process $p$ samples from the distribution with density $r$, with all densities interpreted as *relative to the reference measures* $\pi_{\text{ref}}, \overleftarrow{\pi}_{\text{ref}}, \overrightarrow{\pi}_{\text{ref}}$.[2]

If $p_\theta$ is a diffusion model that satisfies the TB constraint jointly with some reverse process $q$ and target density $q(\mathbf{x}_1)$, then one can take the reference transition kernels $\overrightarrow{\pi}_{\text{ref}}, \overleftarrow{\pi}_{\text{ref}}$ to be $p$ and $q$, respectively. In this case, the TB constraint for a target density $\frac{1}{Z} r(\mathbf{x}_1)$ and forward transition $p_\phi^{\text{post}}$ is

$$\frac{p_\phi^{\text{post}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)}{\overrightarrow{\pi}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_1)} = \frac{\frac{1}{Z} r(\mathbf{x}_1) \cancel{q(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1)}}{\cancel{\overleftarrow{\pi}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_{\Delta t}, \ldots, \mathbf{x}_{1-\Delta t} \mid \mathbf{x}_1)}}, \tag{12}$$

[2] Recall that the relative density (or Radon-Nikodym derivative) of a distribution with density $p$ under the Lebesgue measure relative to one with density $\pi$ is simply the ratio of densities $p/\pi$.

which is identical to the RTB constraint (9). If (12) holds, then $p_\phi^{\text{post}}$ samples from the distribution with density $\frac{1}{Z}r(\mathbf{x}_1)$ relative to $\pi_{\text{ref}}(\mathbf{x}_1)$, which is exactly $\frac{1}{Z}p_\theta(\mathbf{x}_1)r(\mathbf{x}_1)$. We have thus recovered RTB as a case of TB for non-Lebesgue reference measures.

## E Other related work

**Composing iterative generative processes.** Beyond the approximate posterior sampling algorithms and application-specific techniques discussed in §1 and §3, several recent works have explored the use of hierarchical models, such as diffusion models, as modular components in generative processes. Diffusion models can be used to sample product distributions to induce compositional structure in images [34, 12]. Amortized Bayesian inference [31, 44, 43, 19] is another domain of sampling from product distributions where diffusion models are now being used [20]. Beyond product models, [18] studies ways to amortize other kinds of compositions of hierarchical processes, including diffusion models, while [50] proposes methods to sample the product of many iterative processes in application to federated learning. Finally, models without hierarchical structure, such as normalizing flows, have been used to amortize intractable inference in pretrained diffusion models [*e.g.*, 16]. In contrast, our method performs posterior inference by *fine-tuning* a prior model, developing a direction on flexible extraction of information from large pretrained models [26].

**Diffusion samplers.** Several prior works seek to amortize MCMC sampling from unnormalized densities by training diffusion models for efficient mode-mixing [5, 64, 57, 46, 58, 2]. Our work is most closely related to continuous GFlowNets [32], which offer an alternative perspective on training diffusion samplers using off-policy flow consistency objectives [32, 63, 49].

## F Posterior inference on two-dimensional Gaussian mixture model

**Setup** We conduct toy experiments in low-dimensional spaces using samples from a Gaussian mixture model with multiple modes to visually demonstrate its validity. The prior distribution $p(\mathbf{x}_1)$ is trained on a Gaussian mixture model with 25 evenly weighted modes, while the target posterior $p^{\text{post}}(\mathbf{x}_1) = r(\mathbf{x}_1)p(\mathbf{x}_1)$ uses a reward $r(\mathbf{x}_1)$ to select and re-weight 9 modes from $p(\mathbf{x}_1)$. More specifically, the resulting posterior is:

$$p^{\text{post}}(\mathbf{x}_1) = \frac{1}{\sum_j \tilde{\pi}_j} \sum_i \tilde{\pi}_i \mathcal{N}(\mathbf{x}_1 \mid \mu_i, \mathbf{I}) \tag{13}$$

$$\{\mu_i\} = \{(-10,-5), (-5,-10), (-5,0), (10,-5), (0,0), (0,5), (5,-5), (5,0), (5,10)\} \tag{14}$$

$$\{\tilde{\pi}_i\} = \{4, 10, 4, 5, 10, 5, 4, 15, 4\} \tag{15}$$

Our objective is to sample from the posterior $p^{\text{post}}(\mathbf{x}_1)$. We compare our method with several baselines, including policy gradient reinforcement learning (RL) with KL constraint and classifier-guided diffusion models. For RL, we implemented the REINFORCE method with a mean baseline and a KL constraint, following recent work training diffusion models to optimize a reward function [6]. Sampling according to the RL policy leads to a distribution $q_\theta(\mathbf{x}_1)$, which is trained with the objective:

$$J(\theta) = \mathbb{E}_{q_\theta(\mathbf{x}_1)}\left[r(\mathbf{x}_1)\right] + \alpha D_{\text{KL}}\left(q_\theta(\mathbf{x}_1) \| p(\mathbf{x}_1)\right) \tag{16}$$

While the exact computation of $KL\left(q_\theta(\mathbf{x}_1) \| p(\mathbf{x}_1)\right)$ is intractable, we follow the approximation method introduced by Fan et al. [15], which sums the divergence at every diffusion step. This approximation optimizes an upper bound of the marginal KL.

The other baseline is classifier (energy) guidance, which given a diffusion prior, samples using a posterior score function estimate:

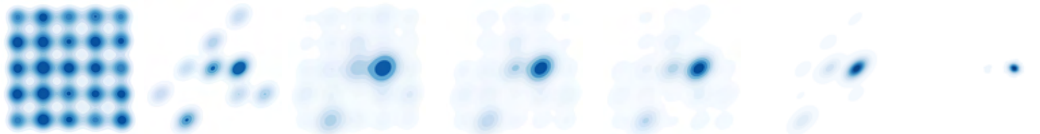
$$\nabla_{\mathbf{x}_t} \log p^{\text{post}}(\mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log r(\mathbf{x}_t) \tag{17}$$

Note that this is a biased approximation of the true intractable score:

$$\nabla_{\mathbf{x}_t} \log p^{\text{post}}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log \mathbb{E}_{p(\mathbf{x}_1 \mid \mathbf{x}_t)}[r(\mathbf{x}_1)] \text{ [36]} \tag{18}$$

For our experiments, we follow the source code[3] provided in recent diffusion sampler benchmarks [49]. We utilize a batch size of 500, with finetuning at 5,000 training iterations, a learning rate of 0.0001,

[3] `https://github.com/GFNOrg/gfn-diffusion`



(a) Prior (b) Posterior (c) $\alpha = 1.0$ (d) $\alpha = 0.8$ (e) $\alpha = 0.7$ (f) $\alpha = 0.5$ (g) $\alpha = 0.3$

Figure F.1: Tuning the KL weight $\alpha$ in reinforcement learning: influences the balance between sticking to the prior distribution and moving towards the modes of the reward density. A higher $\alpha$ value maintains closer adherence to the prior, while a lower $\alpha$ allows a gradual shift towards high values of $r(\mathbf{x})$. Setting $\alpha$ below 0.3 tends to cause mode collapse, moving too far from the prior and focusing on maximizing rewards for single modes. $\alpha = 0.5$ gives us samples that closest resembles the posterior.

(a) Prior (b) Posterior (c) RTB (on) (d) RTB (off) (e) RL (on) (f) RL (off)

Figure F.2: Off-policy exploration benefits for RTB training. RTB, with simple off-policy exploration techniques that increase randomness in the diffusion process, significantly improves mode coverage. On the other hand, policy gradient RL methods which are typically used to finetune diffusion models are on-policy, and hence prone to mode collapse.

a diffusion time scale of 5.0, 100 steps, and a log variance range of 4.0. The neural architecture employed is identical to that used in [49]. For pretraining the prior model, we use the same hyperparameters as above, but with 10,000 training iterations using maximum likelihood estimation with true samples.

**Results.** As we reported in the main text, in Fig. 1, we present illustrative results. The classifier-guided diffusion model shows biased posterior sampling (Fig. 1f), failing to provide accurate inference. RL with a per step KL constraint cannot exactly optimize for the posterior distribution, making the tuning of the KL weight $\alpha$ crucial to achieving desirable output Fig. F.1. RTB asymptotically achieves the true posterior without introducing a balancing hyperparameter $\alpha$. Another advantage of our approach is off-policy exploration for efficient mode coverage. RL methods for fine-tuning diffusion models (*e.g.*, DPOK [15], DDPO [6]) typically use policy gradient style methods that are on-policy. By using a simple off-policy trick introduced by [38, 32] and demonstrated by Sendera et al. [49], we can introduce randomness into the exploration process in diffusion by adding $\frac{\epsilon^2}{T}$, where $\epsilon$ is a noise hyperparameter and $T$ is the diffusion timestep, into the variances and annealing it to zero over training iterations. We set $\epsilon = 0.5$ for off-policy exploration. As shown in Fig. F.2, RTB with off-policy exploration gives very close posterior inferences, whereas off-policy exploration in RL with $\alpha = 0.5$ (which is a carefully selected hyperparameter) does not improve performance due to its on-policy nature.

## G On classifier guidance and RTB posterior sampling

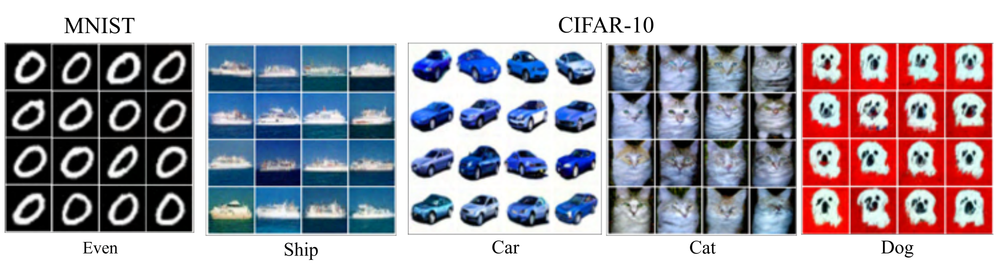



Figure G.1: Samples from a posterior model fine-tuned with RL (no KL). We observe early mode collapse, showcasing high-reward samples with minimal diversity.

## G.1 Experimental Details

In our experiments, we fine-tune pretrained unconditional diffusion models with our RTB objective, to sample from a posterior distribution in the form $p^{\text{post}}(x \mid y) = p(y \mid x)p(x)$. In this section, we detail the experimental settings for RTB as well as the compared baselines.

**Experiments setting.** For MNIST, we pretrain a noise-predicting diffusion model on $28 \times 28$ (upscaled to $32 \times 32$) single channel images of digits from the MNIST datasets. We discretize the forward and backward processes into 200 steps and train our model until convergence. For CIFAR-10, we use a pretrained model from [24], trained to generate $32 \times 32$ 3-channel images from the CIFAR-10 dataset, while discretizing the noising/denoising processes into 1000 steps. For fine-tuning the prior, we parametrize the posterior with LoRA weights [25], with the number of parameters equal to about 3% of the prior model's parameter count. We train our models on a single NVIDA V100 GPU.

We compute FID as a similarity score estimate of the *true* posterior distribution from the data. As such, the computation is limited to the total number of per-class-samples present in the data, (between 5k and 6k for CIFAR-10 and MNIST digits, and 30k for the even/odd task).

**RTB.** For RTB fine-tuning, we finetune a diffusion model following the objective in Equation 1. We impose the objective while sampling denoising paths following a DDPM sampling scheme, with only 20% to 50% of the original trained steps. We employ loss clipping at 0.1, to account for imperfect constraints in the pretrained prior, and train each of our models for 1500 training iterations, well into convergence trends.

**RL [15].** We implement two RL-based fine-tuning techniques derived from DPOK [15] and DDPO [6], respectively with and without KL regularization. By following the same sampling scheme as in our RTB experiments, we enable a direct comparison with RTB. To fine-tune the KL weight, we perform a search over $\alpha \in \{0.01, 0.1, 1.0\}$.

**DP [8].** We implement and adapt the Gaussian version of the posterior sampling scheme in [8], originally devised for noisy inverse problems. This method relaxes some of our experimental constraints, as it requires a differentiable reward $r(\mathbf{x})$. We perform a sweep over ten values of the suggest parameter range for the step size $\zeta \in [.1, 1.]$ on MNIST single-digit sampling, and choose $\zeta = 0.1$ for our experiments.

**LGD-MC [52].** We adapt the implementation of the algorithm in [52] to sample from the classifier-based posteriors in CIFAR-10 and MNIST. Similarly to the DP baseline, we use our pretrained classifier to perform measurements at each sampling step, and use a Monte Carlo estimate of the gradient correction to guide the denoising process. We choose $\zeta = 0.1$ following the DP experiments and default the number of particles to 10 as per the authors' guidelines.

## G.2 Additional findings.

**Classifier-guidance baselines.** We find that the DP and LGD-MC classifier-guidance based baselines struggle to sample from the true posterior distribution in our experimental settings. The baselines achieve the lowers classifier average rewards in all tested settings. Despite choosing $\zeta = 0.1$ as the validate best performing hyperparameter, we also also observe the posterior samples from DP and LGD-MC to be close to the prior. As such, DP and LGD-MC score high in diversity, and low in FID for the Even/Odd experimental scenario, as expected from prior sampling benchmarks, but failing to appropriately model the posterior distribution.

**RL and mode collapse.** In the pure Reinforcement Learning objective imposed for the experiments in §3.1 (no KL), we observe a significantly higher reward than other baseline methods, while showcasing increased FID and lower diversity. In Fig. G.1 we show a random set of 16 samples for posterior models trained on 4 different classes of the CIFAR-10 datasets, as well as the *Even* objective from the MNIST dataset, after 500 training iterations. In the figure, we observe early mode collapse and reward exploitation, visually evident from the little to no variation amongst samples for each class class, and single-digit collapse in the multi-modal *even* digits objective (see samples in Fig. 2 for comparison with our RTB-finetuned models).