Denoising Uncertainty via Gaussian Distributional Diffusion Models

Noa Margeta[1] **Jes Frellsen**[1,2] **Ignacio Peis**[1,2]

[1]Technical University of Denmark, Copenhagen, Denmark
[2]Pioneer Centre for Artificial Intelligence, Copenhagen, Denmark

Abstract

Diffusion-based generative models achieve state-of-the-art performance across domains such as image synthesis, molecular design, and protein structure modeling. In the infinitesimal-step limit, learning the first moment of the clean-data posterior is equivalent to learning the score and is theoretically sufficient for exact sampling. In practice, however, discretization of the reverse process induces a mismatch that underestimates posterior uncertainty. Whereas prior work has proposed accurate implicit formulations primarily to accelerate sampling, we instead introduce a simpler yet effective Gaussian approximation of the clean-data posterior with a focus on uncertainty quantification. Despite its parametric form, this choice enables closed-form expressions for all relevant distributions, yielding a training procedure nearly identical to standard diffusion models while efficiently capturing uncertainty. We show that this posterior modeling enhances distributional fidelity and enables competitive outlier detection at substantially lower computational cost than existing approaches.

1. Introduction

Diffusion-based generative models (Ho et al., 2020; Karras et al., 2022; Song et al., 2021) are a powerful paradigm in deep generative modeling, with strong performance in image and video synthesis, molecular design, protein generation, and PDE solving (Song et al., 2021; Ho et al., 2022; Hoogeboom et al., 2022; Watson et al., 2023; Huang et al., 2024). These models define a continuous transformation between data and noise through stochastic dynamics, where learning reduces to score estimation in the infinitesimal-step limit (Anderson, 1982; Vincent, 2011; Hyvärinen, 2005).

In practice, diffusion dynamics are discretized. Many models directly parameterize this discretized process using Markovian kernels, yielding a finite sequence of denoising steps (Ho et al., 2020; Song et al., 2022; Lu et al., 2022). From this perspective, standard score-based formulations effectively collapse the posterior of clean data given a noisy observation to a point estimate. This has two consequences: accelerated sampling becomes sensitive to prediction errors, and uncertainty in the denoised signal is not explicitly represented.

Recent work relaxes this assumption by learning richer denoising distributions through implicit methods, such as adversarial training (Xiao et al., 2022) or scoring-rule-based objectives (Bortoli et al., 2025), often with a focus on accelerated sampling. These approaches can capture complex posteriors, but they typically increase implementation and training complexity.

In this work, we focus on explicit and tractable uncertainty modeling. We introduce *Gaussian Distributional Diffusion Models (GDDM)*, which approximate the clean-data posterior with a data-dependent Gaussian distribution. Despite its simple form, this yields closed-form reverse transitions

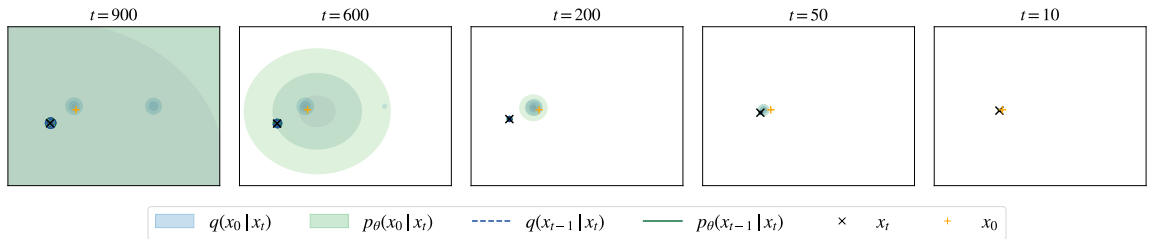


Figure 1. Learned Gaussian posterior $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ on a 2D GMM across noise levels t: blue contours are true posteriors $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$, green contours are model approximations.

and a training objective closely aligned with standard noise prediction. The resulting model captures denoising uncertainty while remaining compatible with accelerated sampling. Our results show improved variational likelihood metrics and competitive low-cost OOD detection performance.

2. Background

A useful starting point for reverse diffusion is the posterior-marginalization identity

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \int q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)\, q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\, d\boldsymbol{x}_0. \tag{1}$$

In words, the reverse transition is an average over clean-data hypotheses $\boldsymbol{x}_0$ weighted by the posterior $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$. Standard diffusion models do not represent this posterior explicitly; instead, they implicitly collapse it to a point estimate, i.e. a delta $\delta\,(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)$ around the predicted clean signal, modeled as $\hat{\boldsymbol{x}}_0 = f_\theta(\boldsymbol{x}_t, t)$. This approximation is exact in the infinitesimal-step limit and usually works well in practice when reverse steps are sufficiently small (Feller, 1949).

Using the Markovian diffusion formulation, we can compute the first factor in Eq. (1), namely the reverse posterior. The forward process is driven by

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}\big(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\big), \tag{2}$$

which yields the tractable marginal conditioned on clean data:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}\big), \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Applying Bayes' rule, the reverse posterior is also Gaussian:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}\big), \tag{4}$$

with parameters given by

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\,\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\boldsymbol{x}_t, \qquad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t. \tag{5}$$

We refer to Appendix A for extended details on the Markovian formulation on diffusion models.

3. Gaussian Distributional Diffusion Models

Here, instead of committing to a deterministic prediction, we explicitly model uncertainty by introducing a conditional Gaussian distribution over the clean data as an approximation to $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$,

$$p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t),\, \boldsymbol{\Sigma}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t)\big), \tag{6}$$

where both mean and diagonal covariance depend on $(\boldsymbol{x}_t, t)$.

3.1. Marginalized reverse transition

We integrate the reversed posterior $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ over the new posterior $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ to obtain an effective reverse transition. First, we express

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(A_t \boldsymbol{x}_0 + B_t \boldsymbol{x}_t, \tilde{\beta}_t I), \tag{7}$$

we we group factors into

$$A_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}, \qquad B_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}. \tag{8}$$

Since both distributions involved in the marginalised reverse transition of our model are Gaussian and $\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0)$ is linear in $\boldsymbol{x}_0$, the integral remains Gaussian:

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)\big), \tag{9}$$

with

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = A_t \boldsymbol{\mu}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t) + B_t \boldsymbol{x}_t, \qquad \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t) = \tilde{\beta}_t I + A_t^2 \boldsymbol{\Sigma}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t). \tag{10}$$

The covariance combines the reverse posterior variance with the propagated uncertainty coming from $p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_t)$.

3.2. Noise-space parameterization

A learned posterior over the clean data $\boldsymbol{x}_0$ naturally induces a posterior over the forward noise variable $\boldsymbol{\varepsilon}$. Using the affine map in Eq. (25), $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ induces

$$p_\theta(\boldsymbol{\varepsilon} \mid \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t)\big), \tag{11}$$

with

$$\boldsymbol{\mu}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_t - \sqrt{\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}\boldsymbol{\mu}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t), \qquad \boldsymbol{\Sigma}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\boldsymbol{\Sigma}_\theta^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t). \tag{12}$$

We directly predict $(\boldsymbol{\mu}_\theta^{(\varepsilon)}, \boldsymbol{\Sigma}_\theta^{(\varepsilon)})$, which is equivalent and convenient in practice.

3.3. Objective

We refer the reader to Appendix A for a full derivation of the objective for the Markovian formulation of diffusion models. In our setting, the per-step KL term between the true reverse posterior and $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ can be computed in closed form. Defining

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{\varepsilon}) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\boldsymbol{\varepsilon}, \tag{13}$$

$$u_t(\boldsymbol{x}_t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t, \qquad C_t = \frac{\beta_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}, \tag{14}$$

we rewrite

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = u_t(\boldsymbol{x}_t) - C_t \boldsymbol{\mu}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t), \qquad \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t) = \tilde{\beta}_t I + C_t^2 \boldsymbol{\Sigma}_\theta^{(\varepsilon)}(\boldsymbol{x}_t, t). \tag{15}$$

The resulting coordinate-wise KL is

$$\mathcal{L}_t = \sum_{i=1}^{d} \frac{1}{2}\Big(\log\big(\tilde{\beta}_t + C_t^2 \Sigma_{\theta,i}^{(\varepsilon)}\big) - \log \tilde{\beta}_t + \frac{C_t^2(\varepsilon_i - \mu_{\varepsilon,i})^2 + \tilde{\beta}_t}{\tilde{\beta}_t + C_t^2 \Sigma_{\theta,i}^{(\varepsilon)}} - 1\Big). \tag{16}$$

As $\boldsymbol{\Sigma}_\theta^{(\varepsilon)} \to 0$, this recovers the standard DDPM weighted MSE objective.

3.4. Uncertainty-based OOD score

The posterior variance can serve as an uncertainty signal for detecting out-of-distribution (OOD) inputs. Given input $\boldsymbol{x}_0$, we diffuse to $\boldsymbol{x}_t$ and use the predicted posterior variance to define

$$u_t(\boldsymbol{x}_t) = \frac{1}{d} \sum_{i=1}^{d} \Sigma_{\theta,i}^{(\boldsymbol{x}_0)}(\boldsymbol{x}_t, t), \tag{17}$$

then average over early timesteps $\mathcal{T}$:

$$s(\boldsymbol{x}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} u_t(\boldsymbol{x}_t). \tag{18}$$

Higher $s(\boldsymbol{x})$ indicates larger posterior uncertainty and potential distributional shift.

4. Related work

Reverse variance and distributional diffusion. DDPM (Ho et al., 2020) fixes reverse variance, while Improved DDPM (Nichol and Dhariwal, 2021) learns it with a hybrid objective. In both cases, variance controls transition stochasticity but is not an explicit posterior over $\boldsymbol{x}_0$. Distributional Diffusion Models (DDMs) (Bortoli et al., 2025) instead learn $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ directly with implicit generators and proper scoring rules (Gneiting and Raftery, 2007). Our method is a Gaussian restriction of this distributional view, preserving posterior modeling while enabling closed-form reverse transitions and KL terms.

Uncertainty and OOD detection. Uncertainty in generative models has been studied through Bayesian parameter inference, ensembles, and variational approximations (Kou et al., 2024; Jazbec et al., 2025; Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016; Kendall and Gal, 2017). For OOD detection, likelihood alone can be unreliable (Nalisnick et al., 2019; Ren et al., 2019), motivating alternative scores (Liu et al., 2021). Recent diffusion-based OOD methods often rely on reconstruction or trajectory-based scores requiring repeated reverse passes (Graham et al., 2023). In contrast, we use the learned posterior variance as a direct early-timestep uncertainty signal, avoiding expensive reconstruction.

5. Experiments

We evaluate GDDM on posterior fidelity, variational bound quality, sample quality, and OOD detection. Unless otherwise stated, models use the same architecture and diffusion schedule.

Code is available at `https://github.com/nmarg/gddm`.

5.1. Posterior recovery in a two-dimensional mixture

We first test whether GDDM approximates $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ in a controlled two-component Gaussian mixture where the true posterior is available in closed form (details in Appendix B). As observed in Fig. 1, at high noise levels, the learned Gaussian posterior substantially overestimates the variance. While this under-confidence is generally undesirable, it also partially mitigates poor mean predictions by assigning lower confidence to displaced estimates rather than committing sharply to the wrong location. Additionally, we include in Appendix C.4 (Fig. 3) the averaged variance per pixel for CIFAR-10.

Table 1. OOD detection performance (AUROC %) for diffusion-based methods (higher is better). Bold indicates the best performance per column. Rank is the average rank across all OOD comparisons (lower is better).

		FashionMNIST			CIFAR-10				Rank
Method	**Model evals**	MNIST	VFlip	HFlip	SVHN	VFlip	HFlip	CelebA	
Reconstruction									
Improved DDPM	2812	**94.11**	**82.56**	63.35	90.13	64.92	50.06	59.05	2.86
Baseline DDPM	2812	93.63	50.11	**63.81**	94.85	**66.16**	**50.58**	60.12	**2.14**
GDDM	2812	92.67	50.28	63.67	96.73	62.75	50.10	57.42	2.86
Uncertainty									
GDDM	4	61.86	43.16	48.25	**96.77**	48.47	50.03	**78.43**	3.86
GDDM	1	79.62	45.78	49.04	97.20	48.91	49.88	80.28	3.14

5.2. Variational bound on CIFAR-10

We evaluate negative variational bound in bits per dimension (BPD) on CIFAR-10. Table 2 shows that GDDM achieves lower BPD than DDPM and Improved DDPM, indicating a tighter bound and better match to the reverse posterior.

5.3. Sample quality and accelerated sampling

We report Fréchet Inception Distance (FID) under full and accelerated sampling. Table 2 shows that baselines obtain better full-step FID, while our method remains competitive under deterministic accelerated sampling and is more sensitive under stochastic acceleration. These results are consistent with the likelihood–perceptual quality trade-off.

Table 2. ELBO evaluation in bits per dimension on CIFAR-10 test set (lower is better).

Model	**BPD**	**FID**
DDPM (baseline)	4.5344	5.53
Improved DDPM	4.2722	4.54
GDDM (ours)	**4.2442**	8.71

5.4. Out-of-distribution detection

See Appendix C.2 (Figure 2) for uncertainty-gap curves over diffusion time. The curves show clear ID/OOD separation at early timesteps, with convergence as t increases. Table 1 reports competitive AUROC on strong shifts. Compared with reconstruction-based diffusion detectors, our score uses only $|\mathcal{T}|$ forward evaluations and no reverse trajectories, reducing computation by more than an order of magnitude in our setup. Performance degrades on subtle semantic shifts (e.g., flipped images), indicating that posterior variance alone is not sufficient in all cases.

6. Conclusion

We introduced Gaussian Distributional Diffusion Model (GDDM), which models $p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ as a diagonal Gaussian. This enables closed-form marginalization of reverse transitions and a tractable per-step KL objective equivalent to a heteroscedastic Gaussian likelihood in noise space.

Empirically, the method improves variational bounds over DDPM baselines and yields interpretable uncertainty estimates. In a controlled 2D setting, it captures posterior contraction but not multimodality; on image data, improved likelihood does not necessarily imply better FID. Finally, the learned posterior variance provides an efficient OOD signal at substantially lower cost than reconstruction-based diffusion detectors.

Acknowledgements

This research was supported by the Villum Foundation through the Synergy project number 50091, by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (MLLS, grant no. NNF20OC0062606), and by the Independent Research Fund Denmark (grant no. 5334-00122B and 5334-00076B). Jes Frellsen was further supported by funding from the Reinholdt W. Jorck og Hustrus Fond. Ignacio Peis acknowledges support by Danish Data Science Academy, which is funded by the Novo Nordisk Foundation (NNF21SA0069429).

References

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

Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

Valentin De Bortoli, Alexandre Galashov, J. Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules, 2025. URL https://arxiv.org/abs/2502.02483.

W Feller. On the theory of stochastic processes, with particular reference to applications. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, page 403. Univ of California Press, 1949.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Tilmann Gneiting and Adrian Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

Mark S. Graham, Walter H. L. Pinaya, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Denoising diffusion models for out-of-distribution detection, 2023. URL https://arxiv.org/abs/2211.07740.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Emiel Hoogeboom et al. Equivariant diffusion for molecule generation in 3d. *ICML*, 2022.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37:130291–130323, 2024.

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

Metod Jazbec, Eliot Wong-Toi, Guoxuan Xia, Dan Zhang, Eric Nalisnick, and Stephan Mandt. Generative uncertainty in diffusion models. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 1837–1858. PMLR, 21–25 Jul 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. URL https://arxiv.org/abs/1703.04977.

Siqi Kou, Lei Gan, Dequan Wang, Chongxuan Li, and Zhijie Deng. Bayesdiff: Estimating pixel-wise uncertainty in diffusion via bayesian inference. In *The Twelfth International Conference on Learning Representations*, 2024.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL https://arxiv.org/abs/1612.01474.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2021. URL https://arxiv.org/abs/2010.03759.

Cheng Lu et al. Dpm-solver: A fast ode solver for diffusion probabilistic models. *NeurIPS*, 2022.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality, 2019. URL https://arxiv.org/abs/1906.02994.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2102.09672.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019. URL https://arxiv.org/abs/1906.02845.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, et al. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.

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

Joseph L. Watson et al. Diffusion models for protein structure generation. *Nature*, 2023.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JprM0p-q0Co.

A. Background: Denoising Diffusion Probabilistic Models

A.1. Generative model

Let $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$ denote a clean data sample drawn from the unknown data-generating distribution. Our objective is to learn a parametric model $p_\theta(\boldsymbol{x}_0)$ by minimizing the forward Kullback–Leibler divergence $\mathrm{KL}(p_{\text{data}} \,\|\, p_\theta)$, which is equivalent to maximum likelihood (ML) estimation.

To this end, we define a generative model over a sequence of T latent variables forming a Markov chain that progressively transforms a simple prior distribution into the data distribution. The resulting joint distribution factorizes as

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t), \tag{19}$$

where $p(\boldsymbol{x}_T) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is a fixed prior that initializes the chain, and $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ denotes the *reverse* transition kernel, parameterized by a neural network. The functional form of these transitions is typically chosen to be Gaussian, and remains a modeling choice.

A.2. Training via variational inference

Under the generative model defined above, maximum likelihood training requires marginalizing over all latent variables:

$$\theta^* = \arg\max_\theta \log p_\theta(\boldsymbol{x}_0) = \arg\max_\theta \log \int p_\theta(\boldsymbol{x}_{0:T})\, d\boldsymbol{x}_{1:T}. \tag{20}$$

When the transition kernels $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ are parameterized by neural networks, this marginalization is intractable in general. To obtain a tractable training objective, we adopt amortized variational inference and introduce a variational distribution $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ to approximate the true posterior over latent variables. This yields the following evidence lower bound (ELBO):

$$\log p_\theta(\boldsymbol{x}_0) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} \right] \;=:\; \mathcal{L}(\theta), \tag{21}$$

which provides a valid surrogate objective for maximum likelihood training, analogous to the ELBO used in variational autoencoders (VAEs).

A.3. Forward noising process

The term *diffusion* is inspired by diffusion processes in statistical physics, in which structured signals are gradually transformed into noise. In the current variational formulation, this intuition is formalized by imposing a forward Markov chain that progressively corrupts the data by injecting noise at each step. Starting from a clean sample $\boldsymbol{x}_0$, this *forward* (or noising) process is defined as

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}). \tag{22}$$

Unlike amortized variational inference in models such as VAEs, where the variational distribution is learned and parameterized by neural networks, the forward process in diffusion models is fixed and contains no learnable parameters. A common choice, adopted in DDPMs, is to define the transitions as Gaussian noise injections:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\, \boldsymbol{x}_{t-1},\, (1-\alpha_t) I\big), \qquad t = 1, \ldots, T, \tag{23}$$

where the coefficients $\alpha_t \in (0,1)$ are determined by a predefined *noise schedule*. It is common to parameterize this schedule in terms of noise variances β_t, defined by $\alpha_t \coloneqq 1 - \beta_t$. In practice, the noise schedule is chosen such that β_t is a *monotonically increasing* sequence, while α_t is *monotonically decreasing*. Early timesteps therefore inject only a small amount of noise, whereas later timesteps progressively overwhelm the signal, eventually driving the distribution toward an isotropic Gaussian.

We further define the cumulative product $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t}(1-\beta_s)$, which measures the total signal attenuation after t forward steps.

Importantly, these transitions are linear-Gaussian, which allows many quantities of interest—such as the forward marginals and the reverse posterior—to be computed in closed form. In particular, composing the Gaussian transitions yields an analytic expression for the forward *marginals*, obtained by marginalizing out all intermediate variables:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0,\, (1-\bar{\alpha}_t)I\big). \tag{24}$$

Samples from this distribution can be obtained via the standard reparameterization trick,

$$\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}) = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I). \tag{25}$$

Importantly, when interpreted across time, this same reparameterization underlies the continuous-time probability flow formulation of diffusion models, in which $\boldsymbol{x}_t$ evolves according to a deterministic ODE.

The $\boldsymbol{x}_0 \leftrightarrow (\boldsymbol{x}_t, \boldsymbol{\varepsilon})$ bijection. Equation (25) plays a central role in diffusion models. For fixed t, this relation defines an affine change of variables between $\boldsymbol{x}_0$, $\boldsymbol{x}_t$, and $\boldsymbol{\varepsilon}$:

$$\boldsymbol{x}_0(\boldsymbol{x}_t, \boldsymbol{\varepsilon}) = \frac{1}{\sqrt{\bar{\alpha}_t}}\,\boldsymbol{x}_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{x}_t - \sqrt{\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}\,\boldsymbol{x}_0. \tag{26}$$

This transformation is affine, implying that, conditioned on $\boldsymbol{x}_t$, both $\boldsymbol{x}_0$ and $\boldsymbol{\varepsilon}$ encode the same underlying randomness, expressed in different coordinate systems. This observation plays a key role in our parameterization of the generative model, described in Section 3.2.

Reverse posterior. Another important consequence of the linear-Gaussian forward process is the tractability of the *reverse posterior*. By Bayes' rule,

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}, \tag{27}$$

where we used the Markov property $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$. Since all terms are Gaussian and available in closed form, the resulting posterior is also Gaussian:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\big(\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0),\, \tilde{\beta}_t I\big), \tag{28}$$

with mean and variance given by

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\,\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\boldsymbol{x}_t, \qquad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t. \tag{29}$$

A.4. ELBO decomposition

Expanding the numerator and denominator of Equation (21) using the factorizations of p_θ and q yields a decomposition of the ELBO into a sum of terms associated with individual timesteps:

$$\mathcal{L}(\theta) = \mathbb{E}_q\left[\log p(\boldsymbol{x}_T) + \sum_{t=1}^{T} \log p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) - \sum_{t=1}^{T} \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\right]. \tag{30}$$

This expression can be further rearranged to obtain a form with lower-variance stochastic estimators. Following Appendix A of DDPM (Ho et al., 2020) (see also Section 20.2.3 of Bishop's book (Bishop and Bishop, 2023)), the ELBO can be written as

$$\mathcal{L}(\theta) = \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1) - \sum_{t=2}^{T} \mathrm{KL}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,\|\, p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\big) - \mathrm{KL}\big(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,\|\, p(\boldsymbol{x}_T)\big) \right]. \tag{31}$$

Thus, the training objective consists of a likelihood term together with a sum of KL divergences between the true reverse posterior and the learned reverse transition, taken in expectation over all timesteps.

Computing the full ELBO in Equation (31) at every training iteration would be computationally prohibitive. For high-dimensional data, the number of timesteps T is typically on the order of thousands, which would require thousands of forward and backward passes through the neural network. As shown by Ho et al. (2020), efficient training is nevertheless possible by optimizing randomly sampled terms of the ELBO using stochastic gradient descent.

In practice, all expectations are approximated via unbiased Monte Carlo sampling over timesteps, clean data, and noisy data samples. In the next section, we show that these reverse posteriors admit closed-form expressions.

A.5. Parameterization of the generative model

The ELBO in Equation (31) encourages the learned reverse transitions to match, on average, the true reverse posteriors. A natural modeling choice is therefore to parameterize the generative kernel as a Gaussian distribution:

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{x}_{t-1};\, \mu_\theta(\boldsymbol{x}_t, t),\, \Sigma_\theta(\boldsymbol{x}_t, t)\big), \tag{32}$$

where the timestep t is provided as an additional input to the neural network in order to amortize the full sequence of transitions.

Since the reverse posterior $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is Gaussian and known in closed form, minimizing the corresponding KL divergence amounts to matching its moments. In particular, the KL is minimized when the mean and covariance of $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ coincide with those of the reverse posterior. However, inspecting Equation (5) reveals that, under the DDPM construction, the posterior mean depends on the data through $\boldsymbol{x}_0$, whereas the posterior variance $\tilde{\beta}_t$ depends only on the noise schedule and the timestep.

As a result, it is common to fix the covariance of the generative kernel and learn only the mean:

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{x}_{t-1};\, \mu_\theta(\boldsymbol{x}_t, t),\, \sigma_t^2 I\big). \tag{33}$$

Alternative fixed variances are also commonly used in practice (Nichol and Dhariwal, 2021). A detailed discussion of these design choices can be found in Section 3.2 of DDPM (Ho et al., 2020) and Section 20.2 of Bishop and Bishop (2023).

A.6. Simplified objective

As discussed in the previous section, when the variance of the generative kernel is fixed, optimizing the ELBO reduces to matching the mean of the learned reverse transition to that of the true reverse posterior. More precisely, expanding the KL terms in Equation (31) yields the following objective:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{x}_t} \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) - \mu_\theta(\boldsymbol{x}_t, t) \right\|^2 \right] + C, \tag{34}$$

where C collects terms that do not depend on θ, and we use notation L to denote negative of $\mathcal{L}$, such that, when $\mathcal{L}$ is maximized, L is minimized.

In principle, this formulation suggests training the neural network to directly predict the mean of the reverse posterior. As shown in the next section, however, this objective admits alternative parameterizations that lead to more effective and numerically stable training.

A.7. Predicting the clean data

Recall that the mean of the reverse posterior in Equation (5) depends on both the noisy sample $\boldsymbol{x}_t$ and the underlying clean data $\boldsymbol{x}_0$. By construction, however, the generative model p_θ has access only to $\boldsymbol{x}_t$ at test time. Rather than directly predicting the mean of the reverse transition, a natural alternative is to parameterize the model through an estimate of the clean data,

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t), \tag{35}$$

and to define the generative mean by mimicking the reverse posterior with this estimate:

$$\mu_\theta(\boldsymbol{x}_t, t) = \tilde{\mu}_t\big(\boldsymbol{x}_t,\ \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)\big). \tag{36}$$

Under this parameterization, the objective in Equation (34) simplifies considerably. Substituting the means by the expression in 5 yields

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\frac{1}{2\sigma_t^2}\left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\right)^2 \|\boldsymbol{x}_0 - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)\|^2\right] + C, \tag{37}$$

Thus, in this case, optimizing the ELBO is equivalent to minimizing a weighted mean-squared error between the predicted clean data and the ground-truth sample.

A.8. Noise parameterization

While predicting the clean data $\boldsymbol{x}_0$ provides an effective parameterization of the generative model, the structure of diffusion models allows for an alternative and often more convenient formulation. In particular, by substituting the affine relation $\boldsymbol{x}_0(\boldsymbol{\varepsilon})$ from Equation (26) into the expression for the reverse-posterior mean in Equation (5), the latter can be rewritten as

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0(\boldsymbol{\varepsilon})) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\varepsilon}\right). \tag{38}$$

This observation suggests reparameterizing the clean-data estimator itself in terms of a predicted noise variable,

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{x}_0\big(\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\big) = \frac{1}{\sqrt{\bar{\alpha}_t}}\,\boldsymbol{x}_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\,\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t), \tag{39}$$

which yields the following expression for the generative mean:

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\right). \tag{40}$$

Substituting this parameterization into the simplified objective in Equation (34) leads to

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\frac{\beta_t^2}{2\sigma_t^2\,\alpha_t(1-\bar{\alpha}_t)}\,\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\|^2\right] + C, \tag{41}$$

where C denotes terms independent of θ.

This formulation results in a particularly simple training objective, which is often equivalent—up to a timestep-dependent weighting—to a mean-squared error between the true noise $\boldsymbol{\varepsilon}$ and its prediction $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$.

Weighted noise prediction objective. In practice, the DDPM formulation further simplifies this loss by absorbing the timestep-dependent prefactor into a weighting scheme. In particular, DDPM proposes optimizing a weighted version of the noise prediction objective, where the contribution of each timestep is scaled by a deterministic weight that depends on the noise schedule. A commonly used choice corresponds to setting $\sigma_t^2 = \beta_t$, which yields a loss proportional to

$$\mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\|^2\right], \tag{42}$$

up to a known weighting factor. This weighting can be interpreted as controlling the relative importance of different noise levels during training, and does not affect the optimal solution under sufficient model capacity. In practice, this simplified objective has been shown to provide stable optimization and strong empirical performance, and is therefore widely adopted in diffusion-based generative models.

A.9. Sampling

After training, generation proceeds by sampling from the learned reverse Markov chain defined by the transitions $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$. Starting from a sample $\boldsymbol{x}_T \sim p(\boldsymbol{x}_T) = \mathcal{N}(0, I)$, new data points are generated by iteratively applying the reverse dynamics for $t = T, \ldots, 1$. Under the Gaussian parameterization introduced above, each reverse step can be written as

$$\boldsymbol{x}_{t-1} = \mu_\theta(\boldsymbol{x}_t, t) + \sigma_t\, \boldsymbol{z}, \qquad \boldsymbol{z} \sim \mathcal{N}(0, I), \tag{43}$$

where $\mu_\theta(\boldsymbol{x}_t, t)$ is defined via the noise prediction network and σ_t^2 denotes the (fixed or learned) variance. This procedure defines a stochastic sampling process whose distribution converges to $p_\theta(\boldsymbol{x}_0)$ as the number of steps increases.

While this ancestral sampling procedure is conceptually simple, it requires evaluating the neural network at every timestep, typically on the order of thousands of steps. To address this limitation, alternative sampling schemes have been proposed that trade stochasticity for computational efficiency.

B. Derivation of the 2D toy dataset true posterior

In this appendix we derive the closed-form expression for the posterior distribution $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ used in the toy two-dimensional posterior recovery experiment described in Section 5.1. Because both the data distribution and the forward diffusion process are Gaussian, the posterior can be computed exactly using Bayes' rule.

Model specification. The clean data distribution is a two-component Gaussian mixture,

$$p_0(\boldsymbol{x}_0) = \sum_{k=1}^{2} \pi_k\, \mathcal{N}(\boldsymbol{x}_0 \mid \boldsymbol{\mu}_k, \sigma^2 \mathbf{I}), \qquad \pi_1 = \pi_2 = \tfrac{1}{2},$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^2$ and $\sigma^2\mathbf{I}$ is an isotropic covariance.

The forward diffusion process at timestep t is given by

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\, \boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

which induces the conditional distribution

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\, \boldsymbol{x}_0,\, (1 - \bar{\alpha}_t)\mathbf{I}\big).$$

Marginal likelihood. Conditioned on mixture component k, the marginal distribution of $\boldsymbol{x}_t$ is obtained by integrating out $\boldsymbol{x}_0$:

$$q(\boldsymbol{x}_t \mid k) = \int q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)\, p(\boldsymbol{x}_0 \mid k)\, \mathrm{d}\boldsymbol{x}_0.$$

Since both terms are Gaussian, the result is

$$q(\boldsymbol{x}_t \mid k) = \mathcal{N}\big(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\, \boldsymbol{\mu}_k,\, (1 - \bar{\alpha}_t + \bar{\alpha}_t \sigma^2)\mathbf{I}\big).$$

Posterior mixture weights. Applying Bayes' rule, the posterior probability of component k given $\boldsymbol{x}_t$ is

$$w_k(t) \;\equiv\; q(k \mid \boldsymbol{x}_t) = \frac{\pi_k\, q(\boldsymbol{x}_t \mid k)}{\sum_{j=1}^{2} \pi_j\, q(\boldsymbol{x}_t \mid j)}.$$

Since the mixture weights are uniform,

$$w_k(t) \propto \mathcal{N}\big(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\, \boldsymbol{\mu}_k,\, (1 - \bar{\alpha}_t + \bar{\alpha}_t \sigma^2)\mathbf{I}\big),$$

with normalization $\sum_k w_k(t) = 1$.

Posterior over clean data given k. Conditioned on component k, the posterior distribution $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t, k)$ follows from standard Gaussian conditioning:

$$q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t, k) = \mathcal{N}\big(\boldsymbol{x}_0 \mid \boldsymbol{\nu}_k(t),\, \boldsymbol{\Sigma}(t)\big),$$

where the posterior covariance is

$$\boldsymbol{\Sigma}(t) = \left(\frac{1}{\sigma^2}\mathbf{I} + \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}\mathbf{I}\right)^{-1} = \frac{\sigma^2(1 - \bar{\alpha}_t)}{1 - \bar{\alpha}_t + \sigma^2 \bar{\alpha}_t}\,\mathbf{I},$$

and the posterior mean is

$$\boldsymbol{\nu}_k(t) = \boldsymbol{\Sigma}(t)\left(\frac{1}{\sigma^2}\,\boldsymbol{\mu}_k + \frac{\sqrt{\bar{\alpha}_t}}{1 - \bar{\alpha}_t}\,\boldsymbol{x}_t\right).$$

Final posterior distribution. Combining the component-wise posteriors with the mixture weights yields

$$q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \sum_{k=1}^{2} w_k(t)\, \mathcal{N}\big(\boldsymbol{x}_0 \mid \boldsymbol{\nu}_k(t),\, \boldsymbol{\Sigma}(t)\big).$$

C. Experimental extension

C.1. Experimental details

The GDDM model is trained using the reparameterized loss from Section 3.3. It is implemented using a U-Net backbone taken from the repo of the original DDPM paper (Ho et al., 2020). The network was modified to output both the mean and log-variance of the noise distribution $(\boldsymbol{\mu}_\varepsilon(\boldsymbol{x}_t, t), \log \boldsymbol{\sigma}_\varepsilon^2(\boldsymbol{x}_t, t))$, through a single output head with 6 channels in total, 3 for the mean and 3 for the variance. Timestep embeddings are encoded using sinusoidal position encodings and injected into all residual blocks.

In some experiments the GDDM is compared to a baseline DDPM trained on the MSE loss in the noise space (Ho et al., 2020) or to the Improved DDPM trained on the hybrid loss (Nichol and Dhariwal, 2021). All models use the same network. For the baseline DDPM the network has only 3 output channels for the mean.

Models are trained with the Adam optimizer, an initial learning rate of 10^{-4}, and a linear β_t schedule of $T = 1000$ steps.

Toy experiment implementation For the two-dimensional posterior recovery experiment in Section 5.1, the U-Net backbone is replaced by a lightweight fully connected MLP. The network takes $\boldsymbol{x}_t \in \mathbb{R}^2$ and the diffusion timestep t as input and predicts the mean and log-variance of the noise distribution, using sinusoidal timestep embeddings and SiLU activations. The model is trained with the same heteroscedastic Gaussian loss as in the image based experiments, using the Adam optimizer with a learning rate of 10^{-3} for 200 epochs on 20,000 samples from the two-component Gaussian mixture.

C.2. Uncertainty-gap curves over diffusion time

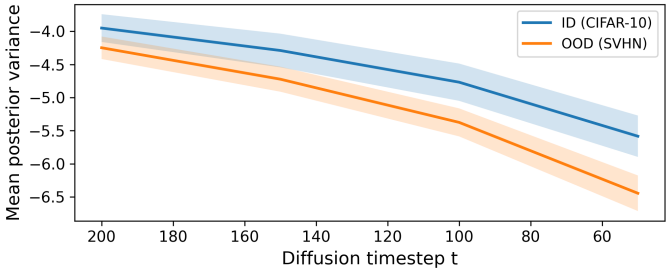


Figure 2. Raw uncertainty trajectories over denoising time, averaged over 500 samples for ID and OOD data.

Figure 2 shows that ID/OOD uncertainty separation is strongest at early reverse steps (low noise), then increases with t as the process becomes noise-dominated. This trend is consistent across dataset pairs and supports the timestep choice used in the ablation table: aggregating uncertainty over early steps preserves the most discriminative OOD signal while keeping computation low.

C.3. OOD uncertainty-based method ablation study

Table 3. Ablation study of the uncertainty-based OOD detection method measured by AUROC. Best results per column are shown in bold. Rank is the average rank across all OOD comparisons (lower is better).

	FashionMNIST			CIFAR-10				
Method	MNIST	VFlip	HFlip	SVHN	VFlip	HFlip	CelebA	Rank
No norm, t=50	0.6270	0.4387	0.4869	**0.9721**	0.4841	0.4985	0.7864	2.43
Z-norm, t=50	0.6270	0.4387	0.4869	**0.9721**	0.4841	0.4985	0.7864	2.43
Z-norm, t=150	0.4108	0.3971	0.4617	0.9103	0.4851	0.4983	0.7240	4.14
Z-norm, t=25	**0.7962**	**0.4578**	**0.4904**	0.9720	**0.4891**	**0.4988**	**0.8028**	**1.29**
Z-norm, t=100–25, Mean	0.6186	0.4316	0.4825	0.9677	0.4847	0.5003	0.7843	2.71
Z-norm, t=100–25, Max	0.5601	0.4282	0.4804	0.9603	0.4813	**0.5004**	0.7876	4.14

We perform an ablation study to analyze the effect of key design choices in the proposed uncertainty based OOD detection method. Specifically, we investigate the impact of: uncertainty normalization, the choice of diffusion timesteps, and the temporal aggregation strategy. Detection performance is measured using AUROC across all ID and OOD dataset pairs described in Section 3.4. The results are summarized in Table 3.

Effect of uncertainty normalization. We compare the use of raw uncertainty values against z-score normalized uncertainty at a fixed timestep ($t = 50$). As shown in Table 3, normalization has

no effect on AUROC across all dataset pairs. This behavior is expected, as AUROC is invariant to monotonic transformations of the anomaly score. Based on this observation, normalization is omitted in subsequent experiments for simplicity.

Effect of timestep selection. We next examine the influence of the diffusion timestep on OOD detection performance. Using uncertainty estimates from early timesteps consistently outperforms using later timesteps across all evaluated datasets. In particular, a substantial degradation in AUROC is observed when only later timesteps are used, indicating that the OOD signal diminishes as the diffusion process approaches higher noise states. This result supports the qualitative analysis in Figure 2 and confirms that uncertainty near the clean data is most informative for OOD detection.

Effect of temporal aggregation. Finally, we compare different temporal aggregation strategies over early timesteps. Averaging uncertainty across timesteps yields consistently higher AUROC than taking the maximum for more challenging semantic shifts such as CIFAR-10 versus CelebA. Mean aggregation provides a more stable estimate of elevated uncertainty while avoiding sensitivity to spurious peaks at individual timesteps.

Summary. Overall, the ablation study indicates that the most effective configuration for uncertainty based OOD detection consists of using raw uncertainty values averaged spatially over pixels and temporally over a small window of early diffusion timesteps. This configuration achieves the best mean rank across all evaluated dataset pairs while maintaining a simple and computationally efficient detection procedure.

C.4. Uncertainty Heatmaps

To further analyze the learned posterior variance, we visualize the diagonal entries of $\Sigma^{(\boldsymbol{x}_0)}_{\theta(x_t,t)}$ across diffusion timesteps. These per-pixel variance maps provide a spatial representation of the model's uncertainty about the clean signal $\boldsymbol{x}_0$.

Normalization Schemes. Because the absolute scale of $\Sigma^{(\boldsymbol{x}_0)}_{\theta(x_t,t)}$ varies significantly across timesteps, we consider two normalization strategies for visualization:

1. **Per-timestep normalization.** For each timestep separately, we rescale variance values to the $[0, 1]$ range:
$$\tilde{\Sigma}_i(x_t, t) = \frac{\Sigma^{(\boldsymbol{x}_0)}_{\theta,i}(\boldsymbol{x}_t, t) - \min_i \Sigma^{(\boldsymbol{x}_0)}_{\theta,i}(\boldsymbol{x}_t, t)}{\max_i \Sigma^{(\boldsymbol{x}_0)}_{\theta,i}(\boldsymbol{x}_t, t) - \min_i \Sigma^{(\boldsymbol{x}_0)}_{\theta,i}(\boldsymbol{x}_t, t)}.$$
This highlights spatial structure within each timestep but removes cross-timestep magnitude comparisons.

2. **Global log-normalization.** We apply a logarithmic transform to the variance values and normalize using global minimum and maximum values across all timesteps:
$$\hat{\Sigma}_i(x_t, t) = \frac{\log \Sigma^{(\boldsymbol{x}_0)}_{\theta,i}(x_t, t) - m_{\min}}{m_{\max} - m_{\min}},$$
where $m_{\min}$ and $m_{\max}$ are computed over the entire trajectory. This normalization preserves relative scale differences across timesteps.

Observations. Figure 3 illustrates representative examples.

Under global normalization, posterior variance decreases monotonically as $t \to 0$, illustrating posterior contraction as the signal-to-noise ratio increases. At high noise levels, uncertainty is spatially

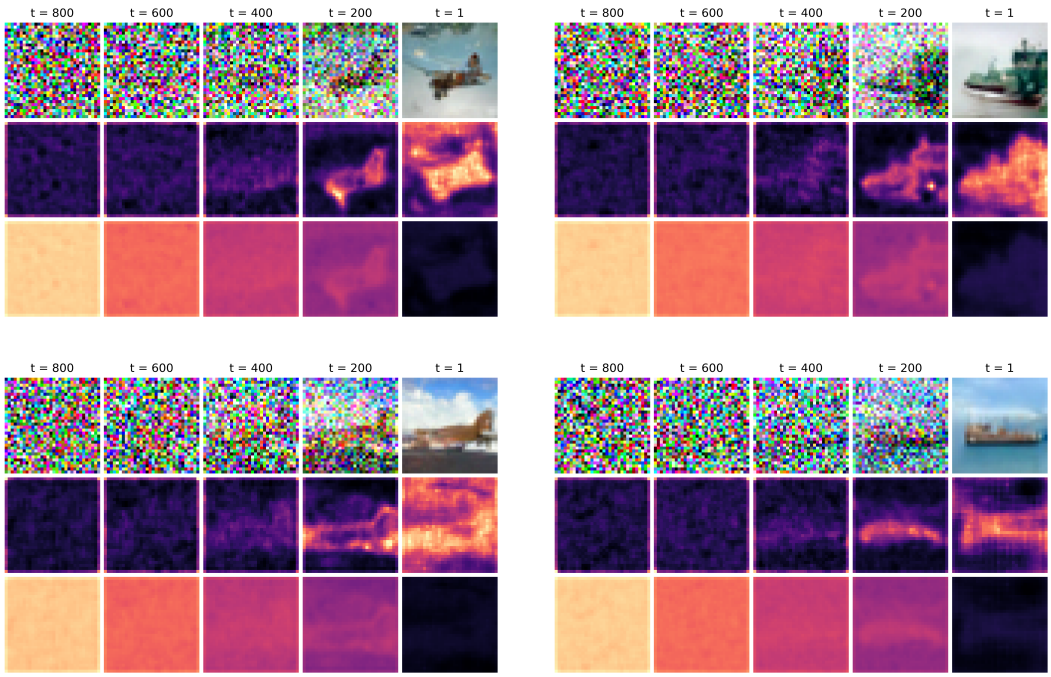


Figure 3. Uncertainty heatmap visualizations for four representative generated samples at selected diffusion timesteps. For each sample, columns correspond to diffusion timesteps. Top row: generated images. Middle row: per-timestep normalized posterior variance highlighting spatial uncertainty structure. Bottom row: globally log-normalized posterior variance illustrating the magnitude of uncertainty across timesteps.

diffuse and approximately uniform. As denoising progresses, variance concentrates around object boundaries, fine textures, and semantically ambiguous regions.

Per-timestep normalization reveals that spatial structure becomes increasingly localized during reverse diffusion. Regions corresponding to edges and complex patterns consistently exhibit elevated variance, suggesting that the model allocates greater uncertainty to areas requiring fine-grained reconstruction.

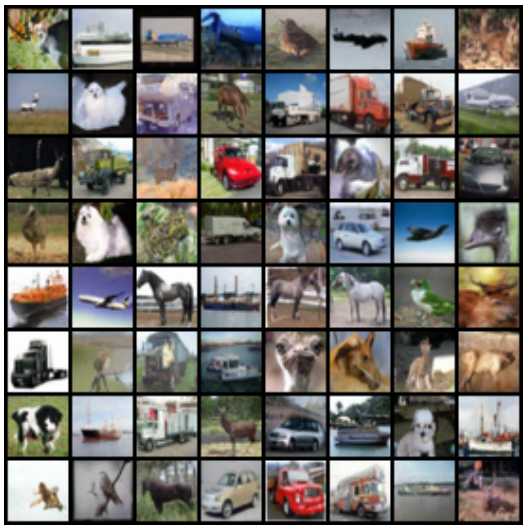
Despite these structured patterns, the Gaussian parameterization enforces unimodality, limiting the ability to represent multiple plausible reconstructions.

D. Sample grids

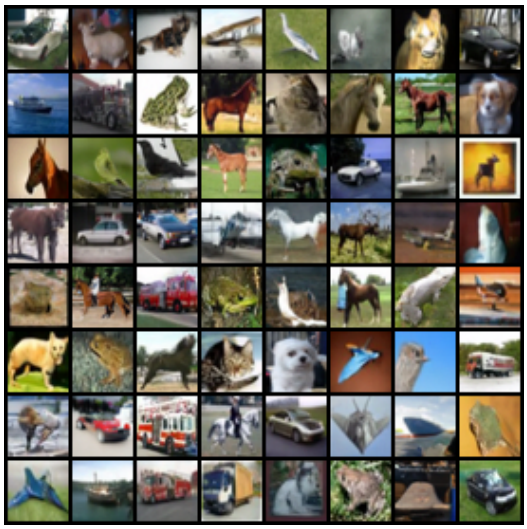
This section provides qualitative comparisons for full-step and accelerated sampling. Figure 4 compares representative full-timestep generations, while Figure 5 shows the effect of reducing the number of reverse steps.

D.1. Full timestep sampling

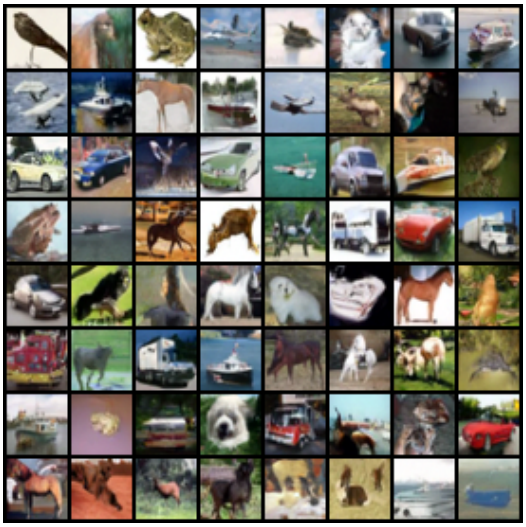
Figure 4 compares DDPM, Improved DDPM, and GDDM under full diffusion trajectories. The samples illustrate that GDDM remains visually competitive while preserving the uncertainty-aware training objective.



(a) DDPM



(b) Improved DDPM



(c) GDDM

Figure 4. Full-timestep sampling comparison between DDPM, Improved DDPM, and GDDM.

D.2. Accelerated sampling

Figure 5 reports deterministic and stochastic accelerated sampling at 10, 50, 100, and 200 steps. As expected, fewer steps increase approximation error, with stochastic updates being more sensitive; nevertheless, GDDM maintains coherent samples across the tested acceleration regimes.

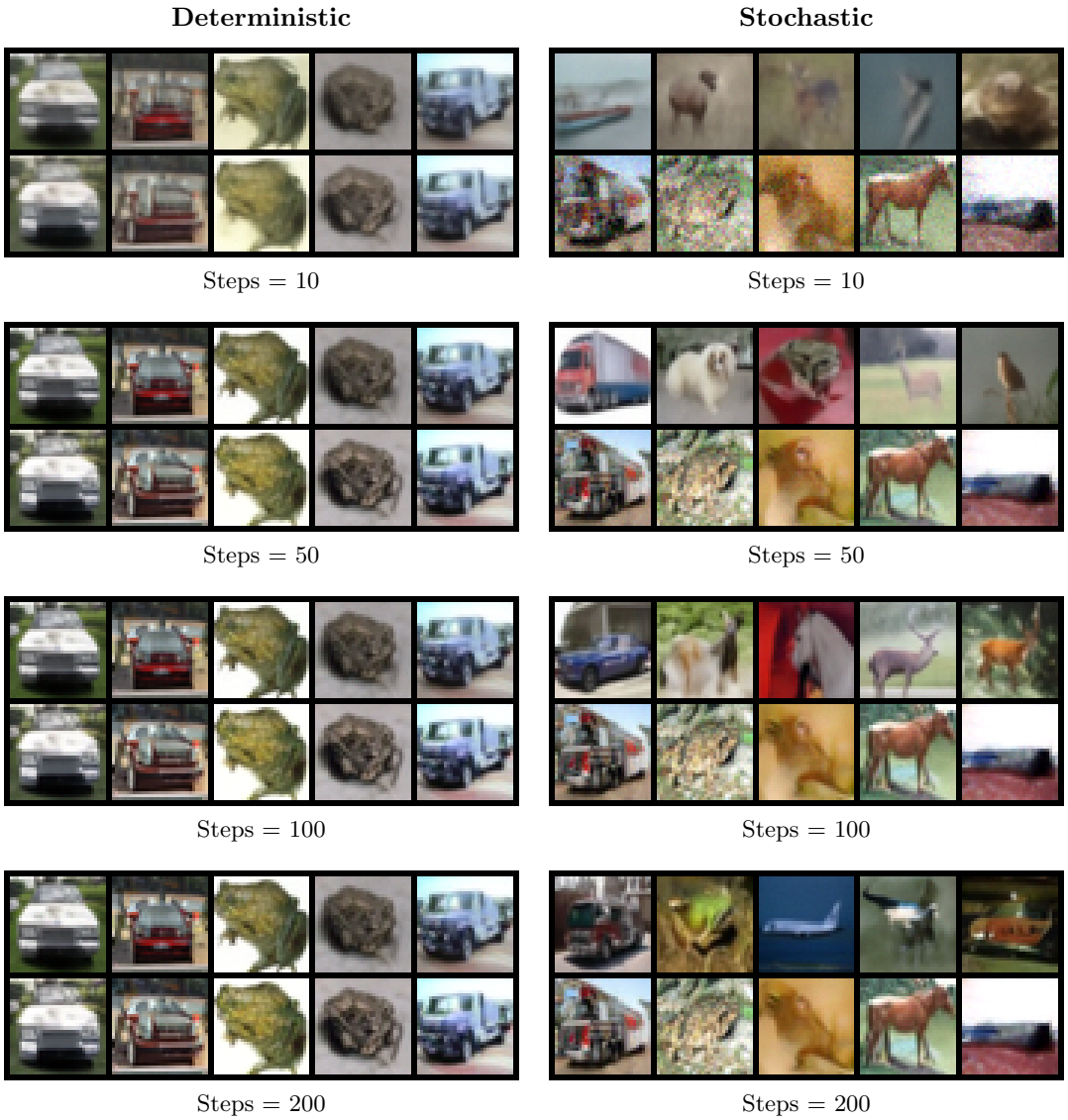


Figure 5. CIFAR-10 samples generated using accelerated sampling at 4 different step counts. Left column shows deterministic accelerated sampling, while the right column shows stochastic accelerated sampling. In each sample grid, the top row corresponds to the DDPM baseline and the bottom row to the GDDM, using the same random seed.