Bernoulli Priors as Efficient Denoising Guides for Diffusion Models

Magdalena Proszewska Nikolay Malkin N. Siddharth
School of Informatics, University of Edinburgh
{m.proszewska,nmalkin,n.siddharth}@ed.ac.uk

Abstract

We draw a connection between representation learning and efficient modelling in the diffusion domain. Particularly we relate the representation-learning capabilities of diffusion autoencoders (DAs) and efficient learning of diffusion models (DMs) that learn their forward process. DAs, through their input-dependent latent variables, can to varying extents be used for representation learning, controllable generation, and interpolation. However, their generative performance relies heavily on how well such a variable can be modelled and subsequently sampled from. DMs with learnable forward processes on the other hand are effective at adjusting the noise scales for the forward process in an input-dependent manner, but this means additional constraints derived from the terminal conditions of diffusion itself. We develop DDPM-BP, a diffusion model that employs a binary *input-dependent variable as a bottleneck to guide the denoising process, satisfying the representation learning capabilities of DAs—evaluated on downstream tasks—as well as efficient learning and generation with fewer denoising steps compared to standard DMs.*

1. Introduction

Diffusion models (DMs) are powerful generative frameworks, with Denoising Diffusion Probabilistic Models (DDPMs) [4] and score-based models [18] as foundations. Despite their generative capabilities, these models are often computationally expensive due to the large number of steps required. To address this, DDIM [17] employs non-Markovian noising for faster sampling, while further refinements improved sample quality and robustness [9, 11, 12].

While most DMs assume a fixed forward noising process and focus on learning the reverse denoising step, recent work has explored addtionally learning the forward noising process itself [1, 10], leading to improved log-likelihoods. These methods allow for more flexible generative trajectories and can reduce generation costs.

Meanwhile, recent research has explored use of DMs in representation learning, focusing on either extracting features from pretrained models [22–27], or using Diffusion Autoencoders (DAs) for unsupervised representation learning [13, 14, 21]. DAs leverage an additional input-dependent variable z to guide the denoising process, which enables reconstruction, controllable generation and interpolations. This not only facilitates representation learning but also reduces the number of denoising steps required for generation. However, the extent of this efficiency gain depends on how well that z can be sampled during inference.

We argue that there is a strong connection between DAs and DMs with learned forward processes. In particular, when the latent variable z can be effectively sampled, the principles behind DAs can be leveraged to boost the performance of standard unconditional DMs, much like the benefits gained from learning the forward process. Our contributions are as follows:

1. We outline the connection between Diffusion Autoencoders and DMs with a learnable forward process.
2. We propose DDPM-BP to leverage an additional input-dependent binary variable z to guide both the denoising and noising processes implicitly.
3. We show that DDPM-BP requires fewer denoising steps to produce high-quality samples and learns meaningful representations, without relying on additional loss terms, constraints on z, or auxiliary samplers.

2. Related work

Diffusion Models with learned forward process: DMs gradually corrupt data into noise through a forward process and learn to reverse this corruption. The original DDPM framework [4] and score-based models [18] established this setup with a Markovian noising process $q(x_t|x_{t-1}, t)$. It was later extended by non-Markovian variants [12, 17] where the input x_0 influences the noising process $q(x_t|x_{t-1}, x_0)$, resulting in fewer steps required for inference. These models assume a fixed forward process and focus solely on learning the reverse denoising process.

Recent work explores parameterizing and learning the forward process as well. VDMs [5], NFDMs [1] and DiffEnc [10] learn both the forward process $q_\theta(x_t|x_0, t)$ and the reverse process $p(x_0|x_t)$, and have been shown to achieve

better log-likelihood, potentially requiring fewer steps for denoising. Other work explore conditional diffusion, and use of data-dependent priors [3] or shifts [28]. This direction parallels the motivations behind hierarchical variational autoencoders (VAEs) [7, 19], which introduce multi-level latent structures to better capture data distributions by learning intermediate representations and more flexible priors.

Diffusion Autoencoders (DAs): These combine the benefits of autoencoding and diffusion modeling with a latent variable that guides denoising, enabling reconstruction.

DiffAE [14] employs an encoder $z = \text{Enc}_\phi(x_0)$ whose output is used at each step of denoising alongside x_t and t. For unconditional generation, a separate DDIM model is trained to approximate the distribution of z.

InfoDiffusion [21], based on InfoVAE [29], uses a probabilistic encoder to maximise MI and align the posterior with a discrete prior of z. For inference, z is sampled from the prior, but an unconditional DDPM is used for the first half of the denoising steps, or DDIM is employed as in DiffAE.

DiffuseVAE [13] combines VAE and DDPM in a two-stage training process. First, a VAE learns latent codes and reconstructions; then, a DDPM denoises $p(x_0|x_t, \widehat{x}_0)$, with $\widehat{x}_0$ as the VAE reconstruction of x_0. For inference, latent codes are sampled from a fitted density estimator.

All these DAs can perform reconstruction, controllable generation, and interpolation. Furthermore, they often require fewer denoising steps compared to standard DMs. We draw a connection to techniques like DDIM's non-Markovian forward process and models that learn the noising trajectory, both of which result in similar improvements. We note that DAs learn the forward process implicitly.

Nevertheless, their generative performance heavily depends on auxiliary samplers for z, as sampling directly from priors is either impossible or inefficient. We argue that if z were a variable that could be freely sampled, it would serve as a powerful signal for generation, enabling more efficient (unconditional) generation with fewer denoising steps.

We propose a Diffusion Autoencoder that imposes no additional constraints on z, allowing the model to learn a useful latent variable that simplifies denoising. By defining z as a compact discrete bottleneck, we constrain the posterior to match the prior, enabling direct sampling. This results in a diffusion model that generates high-quality samples efficiently, requiring fewer steps and no auxiliary samplers.

3. Method

Forward diffusion: Given data $x_0 \sim q(x_0)$, we define a learnable forward noising process q_ϕ to generate $x_{1\ldots T}$ as

$$q_\phi(x_t \mid x_0, t) = q_\phi(x_t \mid \widehat{x}_t, z) \tag{1}$$

where $z \sim q_\phi(z \mid x_0)$ is a latent variable modeled by a neural network, and $\widehat{x}_t$ follows a standard forward diffusion process [4]. We define the latent space as $z \in \{0,1\}^{|z|}$, where each component z_i follows a Bernoulli distribution.

Table 1. FID score comparison of sampling strategies for DDPM-BP on CIFAR-10, computed with 10K generated samples.

T	$z \sim \bullet$	$\lvert z\rvert$ 16	32	64	128
10	$q_\phi(z\|x_0)$	11.85	10.34	9.16	9.16
	Bernoulli	11.88	10.48	15.55	22.20
	PixelSNAIL	11.70	10.97	11.33	14.98
100	$q_\phi(z\|x_0)$	4.56	4.96	4.61	4.46
	Bernoulli	4.79	5.33	9.33	17.23
	PixelSNAIL	4.53	5.21	6.04	9.54

Reverse generative model: The generative model reverses the forward process, resulting in a hierarchical generative framework

$$p(x_0) = \int_x p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{2}$$

$$= \int_z p(z) \int_{\widehat{x}} p(\widehat{x}_T) \prod_{t=1}^{T} p_\theta(\widehat{x}_{t-1}|\widehat{x}_t, z) \tag{3}$$

where the denoising step is modeled as

$$p_\theta(\widehat{x}_{t-1}|\widehat{x}_t, z) = \mathcal{N}(\widehat{x}_{t-1}; \mu_\theta(\widehat{x}_t, t, z), \Sigma_\theta(\widehat{x}_t, t, z)) \tag{4}$$

with μ_θ and Σ_θ learned by a neural network. Equation (2) corresponds to DM with learned forward with diffusion process x_t, while Equation (3) corresponds to Diffusion Autoencoder with diffusion process $\widehat{x}_t$ and latent variable z.

Training: Following [9], we train the network to predict the added noise by optimizing a combination of mean squared error and the variational lower bound. We set the size of z to be small so it acts as a bottleneck and enables efficient sampling during inference. No additional loss terms or constraints are applied to the posterior $q_\phi(z \mid x_0)$.
For further details, please refer to code available at https://github.com/exlab-research/dmz.

4. Experiments

We evaluate our model, DDPM-BP, and compare it against models with a learnable forward diffusion process and Diffusion Autoencoders. Experiments are conducted on CIFAR-10 [6] and CelebA-64 [8], following prior work.

Our models are developed on prior improvements to DDPMs [9, 11]. The encoders $p_\phi(z \mid x_0)$ are defined as a sequence of Conv + BatchNorm + LeakyReLU blocks, followed by Gumbel-Softmax, and are trained jointly with the denoising UNets. See code for further details.

4.1. Sampling z ablation

We explore several strategies for sampling vectors z during inference—critical for generating high-quality outputs

Table 2. Comparison of conditioning strategies for incorporating the additional variable z into the denoising UNet on CIFAR-10.

conditioning on z	train iter.	NLL (BDP)	Acc	FID@10K Bernoulli	FID@10K $q_\phi(z\|x_0)$
along with t	400K	3.18	33.4	6.25	4.44
via Cross-Att.	250K	3.18	39.5	4.79	4.56

as shown in prior DA work. For example, DiffAE [14] and InfoDiffusion [21] utilise an auxiliary DDIM process to model the distribution over z. As an alternative, InfoDiffusion decodes initial steps $x_{T\ldots T/2}$ using an unconditional version of the denoising network, and then samples z from a prior for the remaining steps $x_{T/2\ldots 0}$. Note that, despite the prior being explicitly constrained during training, it cannot be efficiently utilised for sampling at all inference steps. DiffuseVAE [13] also employs auxiliary samplers and fits a density estimator—a Gaussian Mixture Model (GMM)—to the VAE latent space to enable sampling during inference. We consider the following three methods:

1. **Sampling z from data:** For reference, we compute FID scores for $z \sim q_\phi(z \mid x_0)$, where $x_0 \sim \mathcal{D}$ is taken from data. We denote this strategy as $z \sim q_\phi(z|x_0)$.
2. **Bernoulli Prior:** We sample each latent component independently as $z_i \sim \text{Bernoulli}(p = 0.5)$.
3. **Autoregressive Prior (PixelSNAIL):** Inspired by prior work on discrete latent models [15, 20], we fit a PixelSNAIL model [2] over latent codes to enable sampling. We refer to this sampling method as $z \sim$ PixelSNAIL.

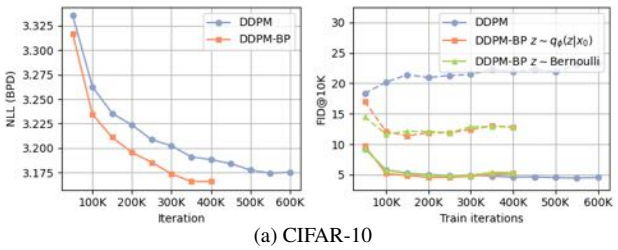
Larger PixelSNAIL models closely match the posterior—or even memorise the dataset—achieving FID scores near those from dataset latents. To ensure a fair comparison, our models are limited to under 600K parameters, based on a grid search that found hyperparameters providing an optimal balance between performance and model size.

FID scores for all strategies are shown in Table 1. Sampling from PixelSNAIL generally yields better results, particularly in higher-dimensional settings. In lower-dimensional latent spaces, the model better leverages the prior, and sampling directly from it yields strong performance without auxiliary samplers. Therefore, we adopt low-dimensional z, optimizing for direct sampling.

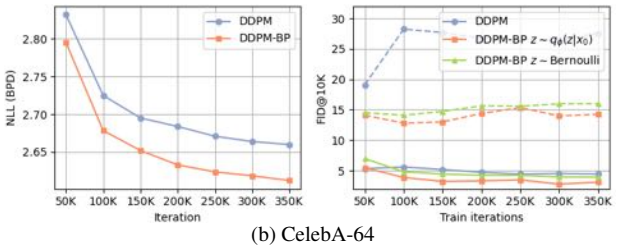
4.2. Denoising network architecture ablation

Both DiffAE [14] and InfoDiffusion [21] condition the denoising UNet on z, along with timestep t, in each block. We explore an alternative approach that replaces standard attention with cross-attention and passing z exclusively through these. This is intended to improve robustness when sampling z from outside the training distribution. A comparison between standard conditioning (using z and t together) and cross-attention using DDPM-BP is shown in Table 2.

The model with cross-attention conditioning trains faster, achieves similar NLL, performs better on the downstream task (Acc), and is more robust with Bernoulli sampling (FID) than the baseline, hence the strategy we adopt.



(a) CIFAR-10



(b) CelebA-64

Figure 1. Comparison of training curves for DDPM-BP and the baseline DDPM. Dashed lines correspond to results for $T = 10$, while solid lines indicate $T = 100$.

4.3. DDPM-BP compared to DDPM

Figure 1 compares training curves of DDPM-BP and the baseline DDPM. As expected, DDPM-BP achieves lower negative log-likelihood (NLL), measured in bits per dimension (BPD), as the latent variable z provides additional context for estimating x_0. For CIFAR-10, it converges faster, while for CelebA, it achieves slightly better FID scores.

Interestingly, we see that for $T = 10$, FID scores worsen over time, particularly for the baseline DDPM. This appears to stem from reduced sample diversity and poor color fidelity, as the generated images tend have a grayish tone. This is notably less prominent in DDPM-BP, suggesting that z helps guide the denoising process more effectively, preserving sample quality even with fewer timesteps.

Moreover, we observe that a lower NLL does not necessarily correspond to better FID scores, highlighting the often-misaligned objectives of likelihood maximization and perceptual sample quality. DMs typically prioritise reducing the number of timesteps T and optimizing NLL or FID—but rarely both. Our work aligns with prior efforts to improve sampling quality and efficiency, not optimize NLL.

4.4. Sampling quality

We assess sample quality using FID scores [16], measuring similarity between the dataset and 10K generated samples. Following prior work [13, 14], we evaluate across varying numbers of inference steps T. Examples of generated images are in Supplementary Material 6. Results are presented in Table 3.

Among all models, VDM achieves the best FID score on CIFAR-10, but requires 1000 denoising steps and evalua-

Table 3. FID scores comparison. All DAs except DiffAE use DDPMs, since DiffAE results are only available for DDIM setting. Models marked * used 50K samples; all others used 10K.

	T	CIFAR-10	CelebA-64
DDPM-BP	10	11.88	15.98
	20	6.92	9.17
	50	5.18	5.13
	100	4.79	3.96
DiffAE* [14]	10	—	12.92
	20	—	10.18
	50	—	7.05
	100	—	5.30
InfoDiffusion [21]	1000	31.5	21.2
DiffuseVAE [13]	10	34.22	25.79
	25	17.36	13.89
	50	11.00	9.09
	100	8.28	7.15
VDM* [5]	1000	4.0	—
DiffEnc [10]	1000	14.6	—
NDFM* [1]	2	12.44	—
	4	7.76	—
	12	5.2	—

tion over 50K samples. For reference, DDPM-BP achieves an FID score of 2.83 on CIFAR-10 for $T = 100$ when using 50K samples instead of 10K. Note that [5, 10] emphasize their focus on optimizing NLL, rather than improving FID scores or reducing the number of inference steps. To our knowledge, [1] is the only work that attempts to optimise for both NLL and FID. While the results reflect this dual optimization, the method is computationally expensive and introduces constraints that limit scalability and integration of other improvements proposed in DMs research [9, 11, 17].

DDPM-BP achieves the best overall results among DAs. Notably, while other models rely on auxiliary samplers to generate latent variables z, we sample z directly from the prior. Additionally, we obtain competitive performance using fewer denoising steps than other approaches.

4.5. Quality of learned representations

We follow the evaluation framework proposed in [21], training logistic regression classifiers using latents z and corresponding labels, and measuring classification performance. We investigate the impact of the size of z on classification scores, as larger latent codes provide more information, making predictions easier. This analysis also helps verify whether the encoder $p_\phi(z|x_0)$ learns meaningful representations. However, as discussed in Section 4.1, we generally opt for smaller $|z|$ for better generative performance. Our results as well as results from [21] are shown in Table 4.

Overall, DDPM-BP yields the best results. Unlike DiffAE, our method learns discrete latent representations, and unlike InfoDiffusion, it does so without relying on auxiliary loss terms to enforce specific structure in the latent space.

(a) CIFAR-10; from top to bottom $|z| = 16, 32, 64, 128$.

(b) CelebA-64; from top to bottom $|z| = 64, 128, 256$.

Figure 2. Samples from DDPM-BPs. Each row contains samples from a different model when using one code $z \sim q_\phi(z|x_0)$.

Table 4. Assessment of learned representation quality based on performance in downstream classification tasks.

	CIFAR-10		CelebA-64	
	$\|z\|$	Acc	$\|z\|$	AUROC
DiffAE	32	39.5	32	79.9
InfoDiffusion	32	41.2	32	84.8
DDPM-BP	16	39.5	64	79.4
	32	41.5	128	80.6
	64	45.6	256	81.0

Figure 2 shows examples of images generated using a single code $z \sim q_\phi(z|x_0)$. As $|z|$ increases, images become more consistent and resemble the reconstruction. For more, refer to Supplementary Material 6.

5. Conclusion

We presented DDPM-BP, a generative model that leverages an additional input-dependent binary latent variable to guide the denoising process. We show connections to efficient learning of DMs by learning the forward process, and conduct experiments and ablations to validate model choices. DDPM-BP requires fewer denoising steps during generation to produce high-quality samples and the learned representations capture meaningful information. Our findings suggest that the use of additional, input-dependent priors provides a compelling and efficient alternative to traditional diffusion modeling.

References

[1] Grigory Bartosh, Dmitry Vetrov, and Christian A. Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling, 2024. 1, 4

[2] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model, 2017. 3

[3] Sang gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior, 2022. 2

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1, 2

[5] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023. 1, 4

[6] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 2

[7] Anna Kuzina and Jakub M. Tomczak. Hierarchical vae with a diffusion-based vampprior, 2024. 2

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. 2

[9] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. 1, 2, 4

[10] Beatrix M. G. Nielsen, Anders Christensen, Andrea Dittadi, and Ole Winther. Diffenc: Variational diffusion with a learned encoder, 2024. 1, 4

[11] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models, 2023. 1, 2, 4

[12] Andrey Okhotin, Dmitry Molchanov, Vladimir Arkhipkin, Grigory Bartosh, Viktor Ohanesian, Aibek Alanov, and Dmitry Vetrov. Star-shaped denoising diffusion probabilistic models, 2023. 1

[13] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents, 2022. 1, 2, 3, 4

[14] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10619–10629, 2022. 1, 2, 3, 4

[15] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 3

[16] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 3

[17] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 1, 4

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 1

[19] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder, 2021. 2

[20] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 3

[21] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. InfoDiffusion: Representation learning using information maximizing diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 36336–36354. PMLR, 2023. 1, 2, 3, 4

[22] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15802–15812, 2023. 1

[23] Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *NeurIPS*, 2023.

[24] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner, 2023.

[25] Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I-Chao Chang, and Hanwang Zhang. Exploring diffusion time-steps for unsupervised representation learning, 2024.

[26] Zijian Zhang, Luping Liu, Zhijie Lin, Yichen Zhu, and Zhou Zhao. Unsupervised discovery of interpretable directions in h-space of pre-trained diffusion models, 2023.

[27] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models, 2023. 1

[28] Zijian Zhang, Zhou Zhao, Jun Yu, and Qi Tian. Shiftddpms: Exploring conditional diffusion models by shifting diffusion trajectories, 2023. 2

[29] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders, 2018. 2

Bernoulli Priors as Efficient Denoising Guides for Diffusion Models

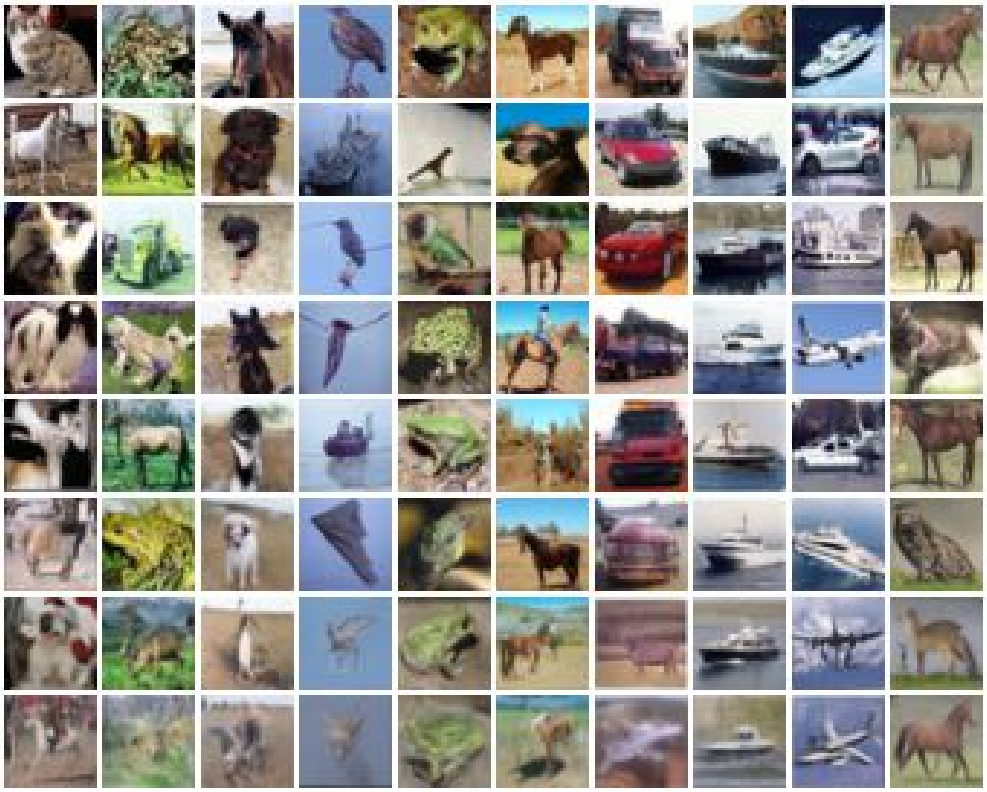
Supplementary Material

6. Samples

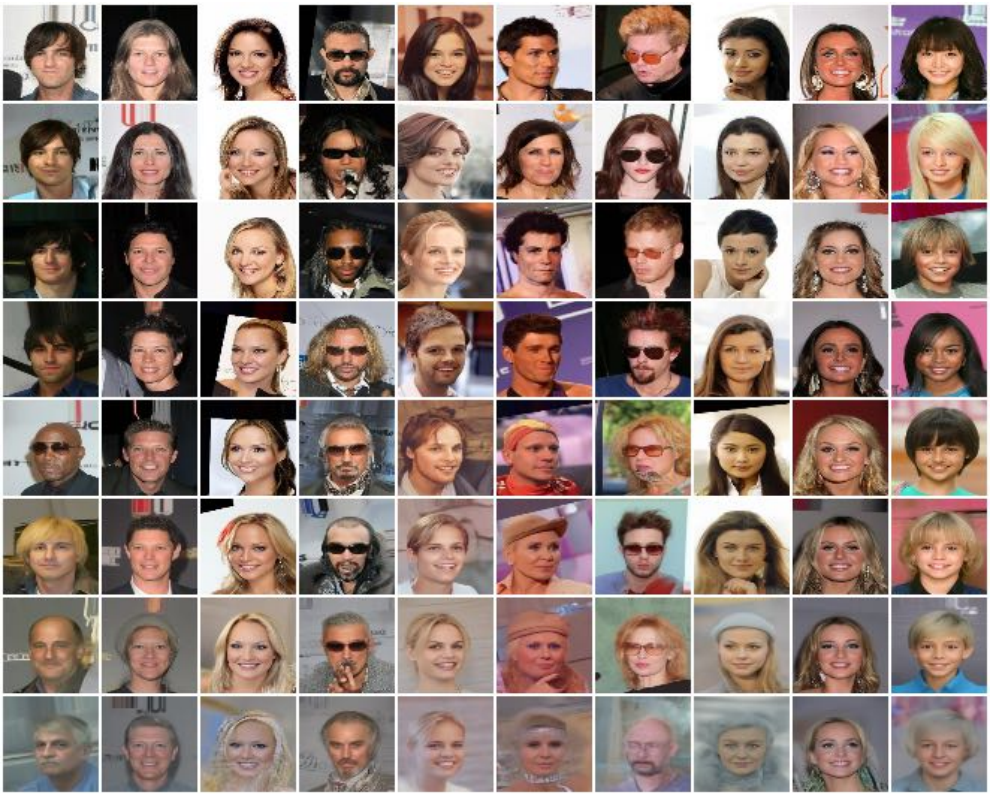
Figure 3 shows examples of images generated with DDPM-BP for different number of denoising steps T. We use DDPM-BP with $|z| = 16$ and $|z| = 64$ for CIFAR-10 and CelebA-64, respectively.

In Figures 4 and 5, we showcase examples of images generated by DDPM-BP to demonstrate both its generative capabilities and the learned representations of z. These samples illustrate how varying the size of z impacts learned representations and the model's ability to reconstruct x_0 from z. This provides insight into how DDPM-BP learns to use the latent space z depending on $|z|$.

In Figure 6, we explore latent space z by generating images using discrete interpolations. For that, we use DDPM-BP with $|z| = 128$ and $|z| = 256$ for CIFAR-10 and CelebA-64, respectively.



(a) CIFAR-10



(b) CelebA-64

Figure 3. Images generated using varying number of denoising steps T. Each column corresponds to one latent code $z \sim$ Bernoulli. Subsequent rows correspond to $T = 1000, 500, 200, 100, 50, 20, 10, 5$.

(a) $|z| = 16$

(b) $|z| = 32$

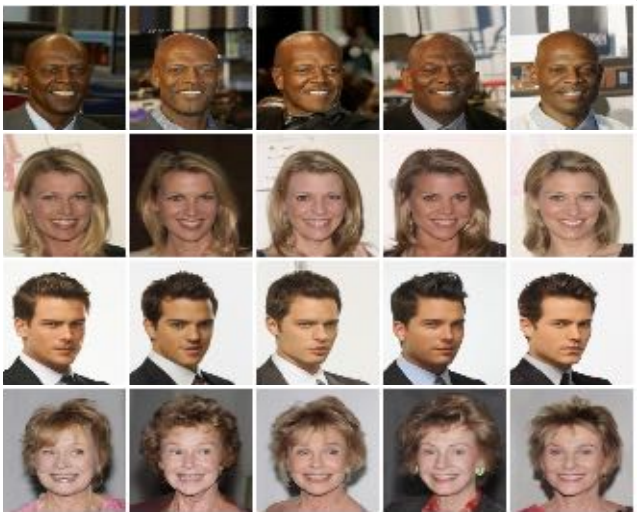
(c) $|z| = 64$

(d) $|z| = 128$

Figure 4. Comparison of CIFAR-10 representations learned by DDPM-BP with varying sizes of z. Images were generated using four different codes $z \sim q_\phi(z \mid x_0), x_0 \sim \mathcal{D}$, and five different $x_T \sim \mathcal{N}(0, \mathbf{I})$.

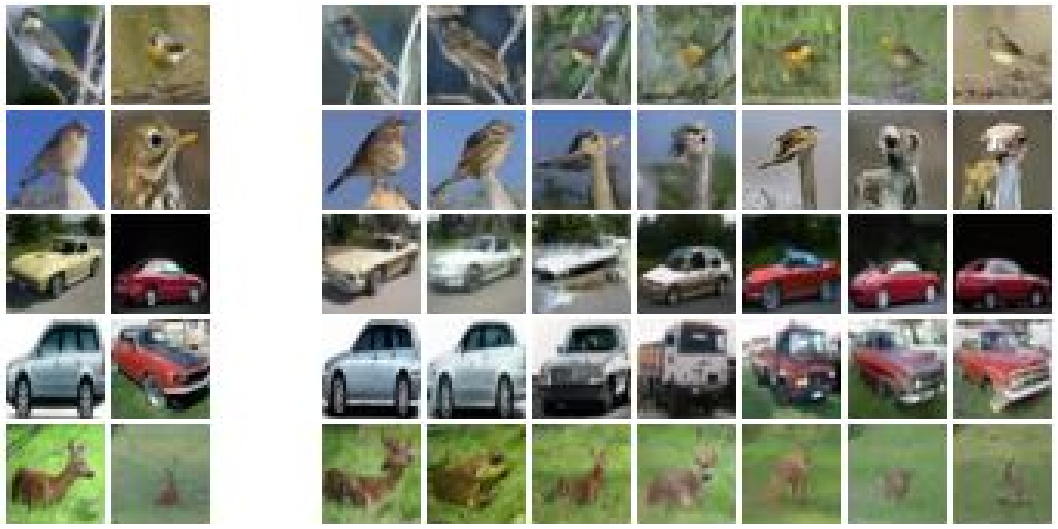
(a) $|z| = 64$

(b) $|z| = 128$



(c) $|z| = 256$

Figure 5. Comparison of CelebA-64 representations learned by DDPM-BP with varying sizes of z. Images were generated using four different codes $z \sim q_\phi(z \mid x_0), x_0 \sim \mathcal{D}$, and five different $x_T \sim \mathcal{N}(0, \mathbf{I})$.



(a) CIFAR-10

(b) CelebA-64

Figure 6. Images generated using discrete interpolations between two codes z^a and z^b, where $z^a \sim q_\phi(z|x_0^a)$, $z^b \sim q_\phi(z|x_0^b)$, $x_0^a, x_0^b \sim \mathcal{D}$, and $x_T \sim N(0, \mathbf{I})$. Original images x_0^a, x_0^b are shown in the first two columns.