On Designing Diffusion Autoencoders for Efficient Generation and Representation Learning

Anonymous authors
Paper under double-blind review

Abstract

Diffusion autoencoders (DAs) are variants of diffusion models that use an input-dependent latent variable to capture representations alongside the diffusion process. The latent variable is typically used for tasks such as downstream classification, controllable generation, and interpolation. However, the latent variable has an effect on unconditional generation, with performance strongly dependent on how this variable is modelled and learnt and how it is used in the denoising process. Here, we draw a connection between DAs and a particular class of improved diffusion models—those that learn their forward (noising) process—to argue that DAs have unrealised potential as generative models. We develop a variant of DAs, termed diffusion models with z (DMZ), to show that the adoption of certain design decisions, such as the choice of latent variables and conditioning method, can help realise this generative potential. Through extensive experiments and ablations, we show that DMZ enables more efficient modelling and generation with fewer denoising steps, as well as providing an effective latent representation for downstream tasks, such as classification and domain transfer.

1 Introduction

Learning effective and efficient deep generative models with latent variables requires satisfying three aims: generating data that matches the observed data distribution, capturing relevant information in latent variables that facilitate interventions or downstream use, and doing both in a computationally efficient manner.

Diffusion models (DMs) are a powerful class of deep generative models that excel at generation, with deep hierarchical models (Ho et al., 2020) and score-based models (Song et al., 2021b) serving as foundations. However, these models are computationally expensive and not designed to capture effective latent representations. Approaches to address efficiency have largely focussed on making generation faster and more robust (Song et al., 2021a; Nichol & Dhariwal, 2021; Ning et al., 2023; Okhotin et al., 2023; Shen et al., 2025). Representation learning approaches using diffusion models have largely focused on extraction from pre-trained models, whether through latent codes (Zhang et al., 2022), internal activations (Yang & Wang, 2023; Xiang et al., 2023), analysis of degradation patterns (Yue et al., 2024), or by aiming to disentangle interpretable structures (Yang et al., 2023; Zhang et al., 2023a).

Diffusion autoencoders (DAs) are a variant of DMs that incorporate input-dependent latent variables to capture representations alongside the diffusion process to enable reconstruction, controllable generation, and interpolations (Preechakul et al., 2022; Wang et al., 2023a). Their effectiveness at capturing such information, ability to subsequently generate data well, and potential do so with fewer denoising steps all depend strongly on how well the latent variable is modelled and fit and how it is used in the denoising process.

Here, we draw a connection between DAs and approaches that learn the forward (noising) process, leading to more efficient learning and better generative models (Kingma et al., 2021; Bartosh et al., 2024; Nielsen et al., 2024), to argue that DAs can be better generative models. We show that certain design decisions, including the choice and dimensionality of latent variable, method of conditioning the denoising process, and setup for the learning the latent distribution allow us to obtain both efficient generation and meaningful latent representations. We term this variant DMZ; i.e., diffusion models with z, to reflect the fact that it only requires a latent variable z with no other components such as regularisers or constraints typical of DAs.

We show that DMZ learns effective representations as evaluated on downstream tasks, including a novel domain-transfer setting for DAs, and permits more efficient learning, modelling, and generation with fewer denoising steps compared to standard DMs. Our contributions are:

1. to show DAs with judicious modelling choices can be efficient diffusion models to train *and* infer with,
2. to explore and ablate design decisions that enable such efficiency, and
3. to evaluate representation quality through their latent variable, on a variety of tasks.

2 Background and related work

Diffusion models (DMs) gradually corrupt data into noise through a forward process and learn to reverse this corruption. Denoising Diffusion Probabilistic Models (DDPMs; Ho et al., 2020) established this foundational setup with a Markovian noising process.

Given sample x_0 from the data distribution $q(x_0)$ and a predefined noise schedule $(\beta_1, \ldots, \beta_T)$, the forward process simulates a Markov chain starting from data $x_0 \sim q(x_0)$, iteratively adding Gaussian noise over T diffusion steps until obtaining a completely noisy image $x_T \sim \mathcal{N}(0, \mathbf{I})$ as

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}),$$
$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right).$$

Given observation x_0, the noised sample at t is derived as $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\alpha_i = 1 - \beta_i$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The reverse process employs a parametric denoiser μ_θ as

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}\right), \quad \text{where}\, \sigma_t = \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}.$$

Instead of directly predicting the mean of the forward process posterior $\mu_\theta(x_t, t)$, Ho et al. (2020) propose training a neural network $\epsilon_\theta(\cdot)$ to predict noise ϵ by optimising

$$\mathbb{E}_{x_0 \sim q(x_0),\, \epsilon \sim \mathcal{N}(0,\mathbf{I}),\, t \sim \mathcal{U}(\{1,\ldots,T\})} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right]. \tag{1}$$

The noise predictor ϵ_θ minimising equation 1 can be expressed in terms of the score of the marginal distribution of x_t, establishing a connection between denoising autoencoders and score-based modelling (Song et al., 2021b). The reverse (denoising) process is then defined in term of μ_θ taken as

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

Markovian DMs were later extended to non-Markovian variants (Song et al., 2021a; Okhotin et al., 2023), where input x_0 influences the denoising process, requiring fewer steps for inference. These models assume a fixed forward process and focus solely on learning the reverse denoising process.

Diffusion models with learned forward process: Recent work explores parametrising and learning the forward process (noising) as well as the denoising process. VDMs (Kingma et al., 2021), NFDMs (Bartosh et al., 2024) and DiffEnc (Nielsen et al., 2024) learn both the forward process $q_\phi(x_t|x_0, t)$ and the reverse process $p_\theta(x_0|x_t)$, and have been shown to achieve better log-likelihood, potentially requiring fewer steps for inference. Other work explores conditional diffusion and use of data-dependent priors (Lee et al., 2022) or shifts (Zhang et al., 2023b). This direction parallels the motivations behind hierarchical variational autoencoders (VAEs) (Vahdat & Kautz, 2020; Kuzina & Tomczak, 2024), which introduce multi-level latent structures to better capture data distributions.

Diffusion autoencoders (DAs): This class of models introduces a latent variable that guides denoising, enabling tasks such as retrieval and editing through learned representations (Preechakul et al., 2022; Wang

et al., 2023a; Pandey et al., 2022; Hudson et al., 2024). Note that DAs are also effectively non-Markovian DMs as they condition denoising on the target x_0 through the latent variable. All instances of DAs excepting Hudson et al. (2024) tackle unconditional generation, which aligns with the focus of our work. DiffAE (Preechakul et al., 2022) employs an encoder $z = \text{Enc}_\phi(x_0)$ whose output is used at each step of denoising alongside x_t and t. InfoDiffusion (Wang et al., 2023a), based on InfoVAE (Zhao et al., 2017), further introduces a probabilistic encoder to maximise MI and align the posterior with a discrete prior of z. DiffuseVAE (Pandey et al., 2022) combines VAE and DDPM in a two-stage process, with a DDPM conditioned on the reconstructions of a pre-trained VAE. While these models demonstrate the ability to control the denoising process via a learned latent variable, they share a key limitation in terms of their generative performance. At inference time, they all rely on auxiliary samplers—such as DDIMs (Preechakul et al., 2022; Wang et al., 2023a) or GMMs (Pandey et al., 2022)—to produce valid latent codes, introducing unnecessary overhead.

3 Diffusion models with z: DMZ

A DA with a stochastic encoder $q_\phi(z \mid x_0)$ is trained with a loss that generalises equation 1:

$$\mathbb{E}_{\substack{x_0 \sim q(x_0),\, \epsilon \sim \mathcal{N}(0, \mathbf{I}), \\ t \sim \mathcal{U}(\{1,\ldots,T\}),\, z \sim q_\phi(z \mid x_0)}} \left[\|\epsilon - \epsilon_\theta(x_t, t, z)\|^2 \right]. \quad (2)$$

Unlike prior approaches that employ additional terms to constrain the latent variables, DMZ directly optimises this objective with respect to the denoiser ϵ_θ (now conditioned on z) and the encoder q_ϕ.

Here, we describe a specific variant of diffusion autoencoders (DAs), following a set of judicious design choices that allow both for efficient generative modelling and capturing effective latent representations. We first draw attention to the key distinctions between DMs, DMs with a learned forward (noising) process, and a particular type of DA, in Figure 1.

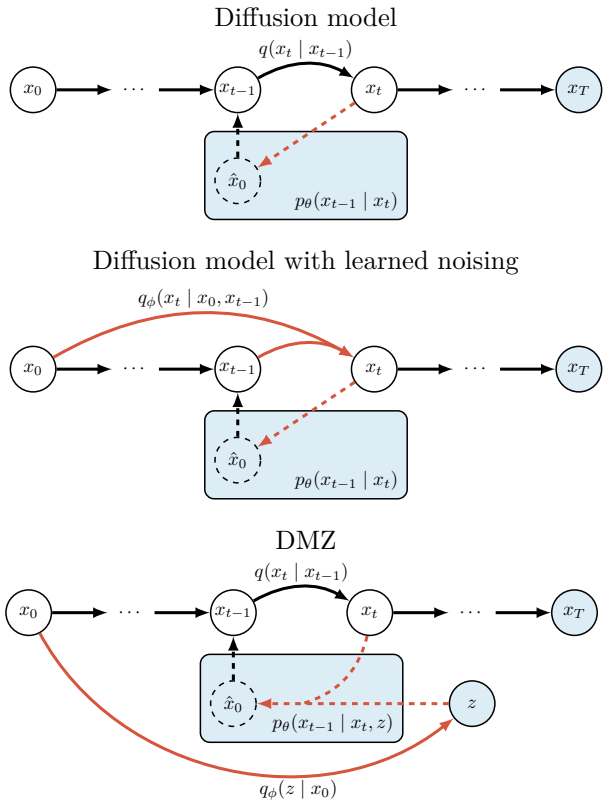


Figure 1: **Top:** Diffusion reverses a Markovian noising process from x_T (possibly via predicted x_0 at each step). **Middle:** A generalisation where generation reverses a non-Markovian learned noising process, marginalising out unknown x_0. **Bottom:** DMZ, where generation conditions on z. Solid and dashed arrows denote noising and generation respectively. Red arrows denote learned parametric models. Blue objects denote data necessary for generation: initial noise, transition kernel, and z.

Diffusion models with a learnable forward process (middle) construct a noised observation x_t by incorporating side-information from the observation x_0, through a learnable parametric function. This means the source for the denoiser, x_t, can carry additional information to help denoise better to x_{t-1}. This is also a fundamental feature of DAs (bottom)—they incorporate side-information through z into the denoising process by additionally conditioning the denoiser on z. In effect, denoising in DAs can be seen as $\{x_t, z\} \mapsto x_{t-1}$, with x_t derived through a standard fixed noising process, just that the information from this x_t and z are not explicitly combined and required to additionally satisfy the constraints of the noising process. This also means that one needs to also be able to sample from the latent z in order to function as a proper generative model; this is what the rest of our design choices seek to address.

Choice of latent z: In choosing the type of latent variables, we note the importance of discrete latent variables for representation learning. They offer a more interpretable and more space-efficient way to represent data compared to continuous latent variables, and can also capture structured relationships and details, leading to simpler and more effective models (Rolfe, 2017; Vahdat et al., 2018; van den Oord et al., 2017;

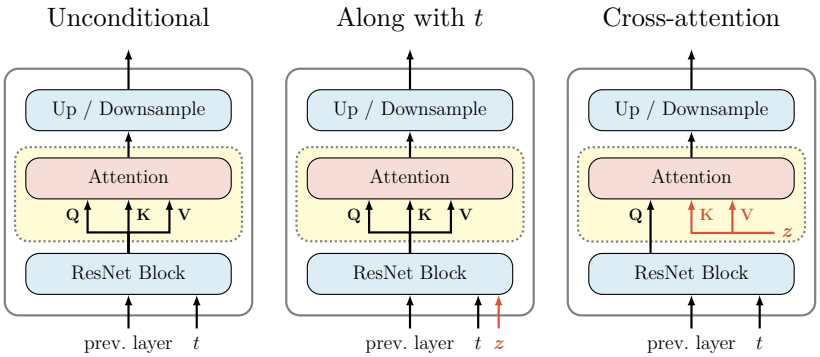


Figure 2: **Left:** Denoiser with time conditioning and optional attention. **Mid, right:** Two conditioning strategies with z.

Razavi et al., 2019; Łukasz Kaiser & Bengio, 2018; Metz et al., 2017). Their type (discrete or continuous) and dimensionality forms the basis of the experiments (Section 4) and ablations (Section 4.5) we perform.

With discrete z, we explore the choice of *binary* latent variables following recent evidence showcasing their effectiveness in the diffusion and reinforcement learning settings Meyer et al. (2024); Wang et al. (2023b). With regard to dimensionality of the latent variable, generally speaking, a lower-dimensional latent is likely to be more interpretable. Conversely, a larger number of dimensions is capable of capturing more information, helping with use in downstream tasks. An additional consideration comes from how the size of the latent variable affects our ability to model a prior over the latent z to allow sampling at inference time.

Conditioning on latent z: Another key design decision for effective modelling is the manner in which the denoiser is conditioned on z. In the standard (DM) case, the denoiser simply takes the noisy observation x_t along with an encoding of time step t. Since existing diffusion autoencoder approaches have been developed and evaluated primarily using U-Net backbones, we adopt the same architecture. A typical model architecture for such conditioning is shown in Figure 2 (top), with the time step given as input to each block of a denoising U-Net, where some blocks include self-attention. A natural way to extend this to condition on z would be to include it along with x_t and t as shown in Figure 2 (middle), as done in some prior DAs (Preechakul et al., 2022; Wang et al., 2023a). However, such global conditioning can entangle the influence of z across all computation paths, potentially encouraging shortcut solutions and reducing selective use of z.

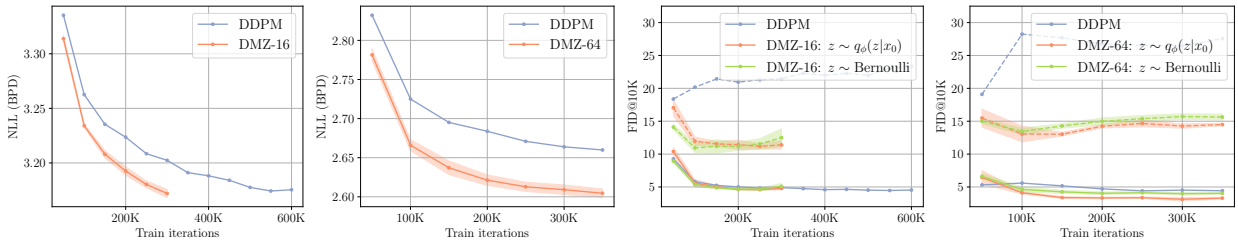
As an alternative, we suggest that z can be more effective when used *only* in modulating attention, and so use *cross-attention* for some blocks with attention layers—with keys K and values V coming from z, and queries Q coming from the original inputs to the denoiser. The rationale is that cross attention allows a parametric function (denoiser) to more flexibly reflect conditioning by transforming the function parameters. Concatenation would only affect conditioning indirectly via the outputs of different layers. As we will see in the experiments, this is a useful inductive bias and can have a marked effect on learning effective and useful representations. While we primarily use a U-Net backbone, the DMZ framework and its cross-attention conditioning strategy extend naturally to Diffusion Transformers (Peebles & Xie, 2022) (see Appendix A.9).

Learning with latent z: An interesting feature of DAs is that the latent variable z is in fact largely redundant in terms of captured information from the data. That is, there is no specific pressure for the model to capture *any* information in z given that the standard noising and denoising processes are sufficiently flexible to faithfully model and generate observed data. This lies in direct contrast to typical deep generative models that employ their latent variables as a *bottleneck*, forcing the flow of all information through them.

However, using z effectively helps better optimise the training objective as it seeks to reconstruct the target observation (or noise). We derive this condition in Appendix A.2, showing that

$$\mathcal{L}_{\text{DMZ}} = \mathbb{E}[\text{Var}(\epsilon \mid x_t, z)] \leq \mathbb{E}[\text{Var}(\epsilon \mid x_t)] = \mathcal{L}_{\text{DDPM}}.$$

This holds for the unconstrained estimator of z. When z is learned via a variational objective, the KL regularisation term penalises information encoded in the latent, making it relatively easy to satisfy constraints



(a) NLL for CIFAR-10 (l) and CelebA-64 (r) (b) FID for CIFAR-10 (l) and CelebA-64 (r)

Figure 3: Comparison of training curves for DMZ and the baseline DDPM. Dashed lines correspond to results for $T = 10$, while solid lines indicate $T = 100$. For DMZ, we plot mean±std over 5 runs.

that encourage $q(z \mid x)$ to match a non-informative prior, such as $\mathcal{N}(0, I)$ (see Appendix A.3). As a result, the model may ignore z if the predictive gain does not offset the KL cost, leading to posterior collapse and a non-informative latent representation. This phenomenon has been observed in prior work (Wang et al., 2023a), where additional mutual information regularisation was used to encourage latent usage. Other approaches avoid this issue by employing pretrained probabilistic models with well-defined priors (Pandey et al., 2022). Therefore, the standard variational regularisation is not well-suited in our setting due to the redundancy of z.

The redundancy of z also means that usual intuitions about the prior's flexibility and capacity do not hold—it only needs to capture information used by the denoiser. As we show experimentally, binary latents suffice. For small latent dimensionality (D), the support size 2^D is comparable to the number of samples drawn from $q_\phi(z \mid x_0)$ across multiple timesteps and data. In this regime, the denoiser encounters nearly all possible latent values during training, so there are few or no unseen configurations at inference time, while still allowing z to encode meaningful information. We thus simply sample from a uniform prior when required at inference time.

Put together, we find that judicious design choices allow us to construct a model that (a) does not need auxiliary losses, (b) does not need additional learning of the prior, (c) captures effective representations, and (d) can do all this while being faster to learn than standard diffusion models.

4 Experiments

We show that DMZ is an efficient and competitive generative model, learns high-quality representations useful for downstream tasks, and extends naturally to a multimodal image-to-image translation framework—all within a unified architecture, for which we provide an ablation study.

All our models are trained following the setup of Nichol & Dhariwal (2021), using their architecture and training hyperparameters. More details can be found in the Appendix and the code[1]. We train until no further improvement in FID scores is observed for $T = 100$ denoising steps. We denote DMZ-n as an instance of DMZ with a latent dimensionality of $|z| = n$. The dimensionality of z was deliberately kept small, guided by the number of available labels in each dataset and the requirements of downstream tasks; this choice is further examined in the ablation study. Additional results and samples are provided in the Appendix.

4.1 Impact of latent z

On training and efficiency: We evaluate the impact of the latent variable z on training efficiency by comparing DMZ to its unconditional counterpart—a standard DDPM. This baseline shares the same architecture and training procedure as DMZ, differing solely in the absence of z-specific components. Following prior work, we train models on CIFAR-10 (Krizhevsky, 2009) and CelebA-64 (Liu et al., 2015). We report FID scores (Seitzer, 2020), computed from 10K samples, and negative log-likelihood (NLL) in bits per dimension.

As shown in Figure 3, DMZ converges faster and achieves better generation efficiency. Notably, for $T = 10$, it achieves much lower FID scores, demonstrating improved performance with fewer denoising steps. Moreover, we observe that a lower NLL does not necessarily correspond to better FID scores, highlighting the often-

[1] https://anonymous.4open.science/r/dmz-9D08

misaligned objectives of likelihood maximisation and perceptual sample quality. We note that our work aligns with prior efforts to improve sampling quality and efficiency in DDPMs, rather than optimising NLL.

On the denoising process: We now examine the role of z at different stages of the denoising process (generation) by measuring the mutual information (MI) between the learned representations $z \sim q_\phi(z \mid x_0)$ and (a) noised data $x_t \sim q(x_t \mid x_0, t)$, and (b) denoised generations $x_t \sim p_\theta(x_t \mid x_{t+1}, z)$, $x_T \sim \mathcal{N}(0, \mathbf{I})$.

For DMZ-16 ($|z| = 16$) trained on CIFAR-10, we fix $T = 100$, and compute MI between sampled z and x_t (t taking 11 evenly spaced values) using MINE (Belghazi et al., 2018)[2]. Figure 4 illustrates how MI evolves over time. We find that while z is theoretically redundant when paired with x_t during training, there is non-negligible MI between them, indicating that the network learns to extract meaningful information from z. Furthermore, z provides information consistently throughout the denoising process.

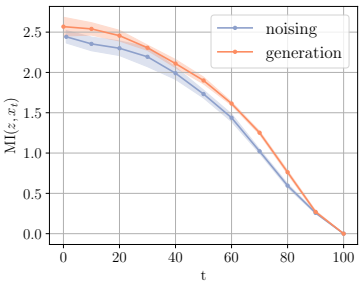


Figure 4: MI between z learned by DMZ and x_t from the noising process (blue) and from the denoising process (orange; starting from $\{x_T, z\}, x_T \sim \mathcal{N}(0, \mathbf{I})$).

4.2 Generation quality and efficiency

We show DMZ is both efficient and competitive, outperforming existing DAs and DMs designed exclusively for generation.

Following prior work, we evaluate DMZ-16 and DMZ-64 on CIFAR-10 and CelebA-64 respectively, using FID scores (FID@50K) across various inference step counts ($T = 10, 20, 50, 100$). Results are presented in Table 1, and include scores obtained by an unconditional counterpart to DMZ, DDPM, which serves as a natural baseline as discussed in Section 4.1. For fair comparison, we also report scores with 10K samples in Appendix A.11. Inception scores are in Table 15.

Table 1: FID scores. All DAs except DiffAE use DDPMs (DiffAE only available as DDIM). Models marked * used 10K samples; all others used 50K. DMZ reports mean and stdev over 5 seeds.

Model	T	CIFAR-10	CelebA-64
DMZ	10	9.78±0.35	14.43±0.45
	20	4.79±0.40	7.89±0.19
	50	3.04±0.28	4.20±0.16
	100	2.79±0.19	2.93±0.10
DDPM (Ning et al., 2023) (reproduced)	10	21.25	25.72
	20	7.21	13.13
	50	3.18	5.69
	100	2.56	3.34
DiffAE (Preechakul et al., 2022)	10	—	12.92
	20	—	10.18
	50	—	7.05
	100	—	5.30
InfoDiffusion* (Wang et al., 2023a)	1000	31.5	21.2
DiffuseVAE* (Pandey et al., 2022)	10	34.22	25.79
	25	17.36	13.89
	50	11.00	9.09
	100	8.28	7.15
VDM (Kingma et al., 2021)	1000	4.0	—
DiffEnc* (Nielsen et al., 2024)	1000	14.6	—
NDFM (Bartosh et al., 2024)	2	12.44	—
	4	7.76	—
	12	5.2	—
DDIM (Song et al., 2021a)	10	13.36	17.33
	20	6.84	13.73
	50	4.67	9.17
	100	4.16	6.53
	1000	4.04	3.51

Following prior work, we evaluate DMZ-16 and DMZ-64 on CIFAR-10 and CelebA-64 respectively, using FID scores (FID@50K) across various inference step counts ($T = 10, 20, 50, 100$). Results are presented in Table 1, and include scores obtained by an unconditional counterpart to DMZ, DDPM, which serves as a natural baseline as discussed in Section 4.1. For fair comparison, we also report scores with 10K samples in Appendix A.11. Inception scores are in Table 15.

DMZ achieves excellent FID scores on both datasets. These scores, even with fewer inference steps, highlight efficiency at generation. Comparing DMZ with the DDPM, we again show (Section 4.1) the benefit provided by including z.

Note that unlike baseline DAs which rely on auxiliary samplers for z, DMZ samples z directly from a Bernoulli prior without additional overhead. DiffAE achieves better FID on the CelebA-64 dataset for $T = 10$, which we attribute to its use of DDIM as the base generative model. We show results combining DMZ with DDIM in Appendix A.4. Overall, DMZ demonstrates stronger performance and is simpler compared to prior DAs.

DMZ achieves performance comparable to state-of-the-art DMs, particularly when accounting for the number of denoising steps required. Additionally, we impose minimal architectural constraints in contrast to NDFM, which introduces constraints that limit scalability and flexibility.

[2]https://github.com/gtegner/mine-pytorch

(a) Latent interpolation using DMZ-256. (b) Targeted edits using DMZ-256 and CelebA attributes.

Figure 6: Qualitative evaluation on CelebA-64.

4.3 Representations

Quantitative evaluation: We assess the quality of the learned representations z following Wang et al. (2023a), by measuring classification accuracy of a logistic classifier trained on the latents. For each dataset, we extract z for all data and report classification performance using 5-fold cross-validation. For CIFAR-10, we report classification accuracy. For CelebA-64, we report the average AUROC over 40 binary attributes to account for class imbalance.

Table 2: Assessment of representation quality via downstream classification tasks. Mean and stdev computed across 5-fold splits.

Dataset →	CIFAR-10		CelebA-64	
Model ↓	$\lvert z\rvert$	Acc	$\lvert z\rvert$	AUROC
DiffAE	32	39.5	32	79.9
InfoDiffusion	32	41.2	32	84.8
DMZ	16	39.50±0.47	32	84.85±0.03
	32	42.35±0.56	64	86.43±0.05
	64	46.45±0.46	128	87.89±0.02
	128	49.71±0.50	256	88.48±0.04

Table 2 presents how $|z|$ impacts classification performance. Our results, alongside those of Wang et al. (2023a), show that DMZ performs on par or better than InfoDiffusion despite having no explicit constraints or regularisation.

DMZ's core idea is to learn representations that guide the denoiser to generate more efficiently. In assessing their effectiveness, while we do not expect state-of-the-art accuracy on downstream tasks, results (Table 2) show the representations are meaningful and compare favourably to our DA baselines. Section 4.4 further explores their utility on a multimodal setup using these representations.

Qualitative evaluation: For each model, we sample data, $x_0 \sim \mathcal{D}$, and its corresponding latent z. For this z, we generate by sampling different noise vectors $x_T \sim \mathcal{N}(0, \mathbf{I})$ and denoise via $p_\theta(x_T, z)$. Figure 5 presents examples illustrating the impact of the size of z on representations. We observe that for smaller $|z|$, less information about the image is retained. Low-level attributes, such as the presence of a smile, are preserved, while higher-level features, such as ethnicity, are not consistently captured. We attribute this to the fact that high-level features remain present in the intermediate representations x_t passed to the network alongside z. As $|z|$ increases, the model is able to encode more information, including higher-level semantic attributes, resulting in generated images that remain consistent across different samples of x_T.

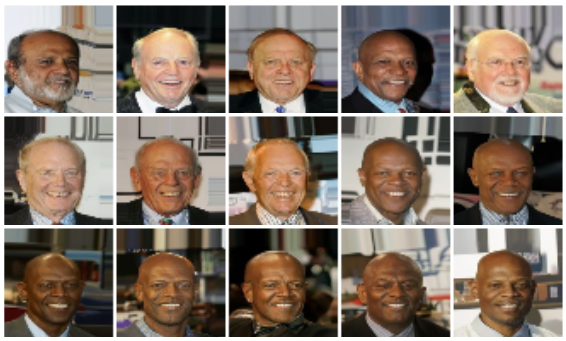


Figure 5: DMZ exemplar generations sharing the same z. Rows correspond to $|z| = 64, 128, 256$.

Next, we illustrate the properties of the learned representations z through interpolation. Specifically, for a given pair of images we extract their corresponding latents, z_{source} and z_{target} and interpolate in discrete steps by sequentially flipping bits in z_{source} to match those in z_{target}. See Algorithm 3 in Appendix A.5 for further details. Figure 6a presents examples for DMZ-256 trained on CelebA-64.

Moreover, via z, we can perform targeted edits on the generated samples—altering attributes like hair or facial expression—using classifiers trained on the latent representations. We use the same classifiers used in the quantitative evaluation edit by moving z along the decision boundary of each classifier. See Algorithm 4 in Appendix A.5 for further details. Examples of images generated by using translations of z with $x_T \sim \mathcal{N}(0, \mathbf{I})$, are shown in Figure 6b.

Figure 7: Exemplar sketch-to-photo translations.

Table 3: Sketch-to-photo translation: edges2handbags.

Model	FID ↓	IS ↑	LPIPS ↓	MSE ↓
Pix2Pix (Isola et al., 2017)	74.8	4.24	0.356	0.209
DDIB (Su et al., 2023)	186.84	2.04	0.869	1.05
SDEdit (Meng et al., 2022)	26.5	3.58	0.271	0.510
Rectified Flow (Liu et al., 2023b)	25.3	2.80	0.241	0.088
I^2SB (Liu et al., 2023a)	7.43	3.40	0.244	0.191
DDBM (VE) (Zhou et al., 2024)	2.93	3.58	0.131	0.013
DDBM (VP) (Zhou et al., 2024)	1.83	3.73	0.142	0.040
DMZ	3.28	3.59	0.359	0.209

4.4 Multimodal framework

We demonstrate how DMZ can be extended to handle multimodal tasks, specifically focusing on image-to-image translation. Inspired by Denoising Diffusion Bridge Models Zhou et al. (2024), we target the sketch-to-photo translation task: edges2handbags (Isola et al., 2017).

Model overview: To effectively reconstruct target (x_0) during translation, the model requires a sufficiently large $|z|$. We train two separate DMZ-512 models: one each for edge images (sketches) and handbags (photos). Training terminates when the mean squared error (MSE) between inputs and samples generated using z shows no further improvement (120K training iterations). The two models learn independent latent spaces—Z_{sketch} for edges and Z_{photo} for handbags. We then learn the mapping between these two latent spaces: $\gamma : Z_{\text{sketch}} \to Z_{\text{photo}}$, where γ is a mapping function parametrised by a multilayer perceptron (MLP).

Sketch-to-photo translation: We follow this pipeline:

(1) **Latent sampling**: We sample a latent variable $z_{\text{sketch}} \sim q_\phi(z|x_{\text{sketch}})$ from the model trained on sketches, where x_{sketch} is the input sketch image.
(2) **Mapping**: We map sampled sketch latent z_{sketch} into the photo latent space using the learned function γ, resulting in latent $z_{\text{photo}} = \gamma(z_{\text{sketch}})$.
(3) **Denoising**: Finally, we use a denoising process to generate the final photo image by sampling $p_\theta(x_T, z_{\text{photo}})$ from the photo model, where $x_T \sim \mathcal{N}(0, \mathbf{I})$.

Results: Following Zhou et al. (2024), we set $T = 40$ and perform sketch-to-photo translations on the training set using our model and baselines. We report FID scores, Inception Scores (IS), LPIPS (Zhang et al., 2018), and Mean Squared Error (MSE) in Table 3. We perform competitively against existing approaches, showing effectiveness of the DMZ framework for multimodal image translation. Crucially, DMZ reduces the reliance on expensive joint training across domains. Additionally, our framework supports unconditional generation of images in both domains (photos and sketches), reverse photo-to-sketch translation, and representation learning. Figure 7 provides examples of the edge-to-handbag translations. Additional examples and discussion are provided in Appendix A.6.

4.5 Ablations

We perform an ablation study to analyse the impact of various design choices on the model's abilities and performance. Specifically, we examine the use of discrete latents, conditioning denoising with z, and the size of the latent variable z. Each of these choices plays a critical role in shaping the model's overall performance and capabilities.

Additionally, we demonstrate that DMZ can be trained via finetuning a pretrained DDPM. While finetuning leads to some loss in efficiency compared to training from scratch, it offers a faster training process, making it an attractive option when time or memory is a limiting factor. Evaluation details and metrics used are as discussed in previous experiments.

Discrete z: First, we explore alternatives to a discrete z, in the form of a Normal prior. We train DMZ-16 on CIFAR-10 using two variants: discrete (Bernoulli prior enforced via Gumbel softmax) and continuous (Normal prior). Table 4 highlights the necessity of using discrete latents. The use of a continuous latent variable makes it infeasible to sample directly from the prior without auxiliary samplers that model the distribution of z. This is evidenced by the poor FID scores obtained when sampling directly from the prior, which, in the continuous version, we determine using the mean and standard deviation of the training data. Additionally, we evaluate a Gaussian Mixture Model (GMM) posterior, as proposed in DiffuseVAE (Pandey et al., 2022). With 10, 20, 50, and 100 mixture components, we obtain FID@50K scores of 5.92, 6.95, 6.07 and 5.47, respectively, which is still way worse than those achieved by the discrete model. Furthermore, the quality of the learned representations declines, as reflected in the lower accuracy on downstream tasks. We attribute this to the latent space becoming more convoluted, making it more difficult for simple logistic classifiers to perform well. We elaborate more on this ablation in Appendix A.7.

Table 4: Performance for discrete vs. continuous latents variables. Normal prior is fit over training data parameters. Comparison uses same number of training steps on CIFAR-10

prior of z	train iter.	NLL (BPD)	Acc	FID@50K ↓		IS@50K ↑	
				prior	$q_\phi(z\|x_0)$	prior	$q_\phi(z\|x_0)$
Normal	250K	3.20	34.62	57.66	3.51	5.76	9.38
Bernoulli	250K	3.18	39.50	2.62	2.79	9.64	9.77

Conditioning via cross-attention: We investigate different strategies for incorporating z into the denoising network. The denoising, unconditional UNets consist of ResNet blocks, up/downsampling blocks, and self-attention blocks. These UNets are conditioned on the timestep t by passing t to each ResNet block. A straightforward way to incorporate z is to provide it alongside the timestep t to each ResNet block (Preechakul et al., 2022; Wang et al., 2023a), e.g., by concatenating their embeddings and using the result in place of the standard timestep embedding (Preechakul et al., 2022). However, we find that incorporating z through cross-attention results in better performance. Figure 2 highlights the architectural differences between the two approaches.

Table 5: Conditioning methods for z with DMZ-16 on CIFAR-10.

method	train iter.	NLL (BPD)	Acc	FID@50K ↓		IS@50K ↑	
				Bern	$q_\phi(z\|x_0)$	Bern	$q_\phi(z\|x_0)$
Along with t	400K	3.18	34.50	7.11	2.42	9.78	9.64
Cross-attention	250K	3.18	39.50	2.62	2.79	9.64	9.77

We replace selected self-attention blocks in the U-Net with cross-attention, enabling better attention over z. This improves robustness to z values not seen during training and leads to better learned representations—reflected in both lower FID scores when sampling from the Bernoulli prior and higher accuracy on downstream tasks (Table 5). Note that while our experiments here are with the U-Net model to ensure fair comparison with other DAs, this cross-attention strategy naturally extends to more recent advances such as Diffusion Transformers (Peebles & Xie, 2022). We include results with DiT architectures in Appendix A.9, noting that the relative gains from including z also hold there.

Small size of z: For an autoencoder to accurately reconstruct x_0 from z, the latent variable z must be sufficiently large. However, the opposite holds true for effective generation: a smaller latent space tends to yield better generative performance (e.g., Yao et al., 2025). Furthermore, a small $|z|$ is sufficient for nearly all use cases. The only exception occurs when reconstruction from z is required, such as in image-to-image translation tasks where Mean Squared Error (MSE) is of concern. Even for image manipulation, a small $|z|$ suffices, as the additional information that a larger $|z|$ could provide is already encoded in the intermediate x_t, which is accessible (unlike in image-to-image translation).

We observe that for larger $|z|$, sampling from the Bernoulli prior becomes less effective. To address this, we explore several strategies for sampling z during inference—a critical component for high-quality generation, as evidenced in prior work on DAs. 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 (van den Oord et al., 2017; Razavi et al., 2019), we fit a PixelSNAIL model (Chen et al., 2018) 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 for latents from data. To ensure fair comparison, our models are limited to <600K parameters, based on a grid search that found hyperparameters which provide an optimal balance between performance and model size.

Table 6: FID@50K scores of sampling strategies for DMZ-16 trained on CIFAR-10.

T	$z \sim \bullet$	$\lvert z \rvert$			
		16	32	64	128
10	$q_\phi(z\|x_0)$	9.70	8.45	7.14	7.34
	Bernoulli	10.03	8.09	13.78	20.57
	PixelSNAIL	9.75	9.08	9.45	12.73
100	$q_\phi(z\|x_0)$	2.79	2.76	2.51	2.45
	Bernoulli	2.62	3.11	7.14	15.09
	PixelSNAIL	2.65	2.87	3.94	7.54

FID scores for all strategies are shown in Table 6. Sampling from PixelSNAIL generally yields better results in higher-dimensional settings. In lower-dimensional settings, the model better leverages the prior, and direct sampling yields strong performance without auxiliary samplers. Therefore, we adopt low-dimensional z, optimising for direct sampling.

Finetuning: All models presented thus far were trained from scratch. Here, we investigate the impact of finetuning strategies on performance. Specifically, when time or computational resources are limited, one might opt to finetune a pretrained DM to accelerate training. We explore how this choice affects the capabilities discussed previously.

We consider three different training strategies: 1) training from scratch, 2) finetuning all parameters, 3) finetuning only the newly added parameters (specifically, those related to the cross-attention mechanism) using a pretrained DDPM. Note the pretrained DDPM used here is unconditional, taken from our previous experiments. Results are in Table 7.

Table 7: DMZ-64: finetuning strategies on CelebA-64.

Finetuning	train iter.	NLL (BPD)	AUROC	FID@50K ↓		IS@50K ↑	
				T=10	T=100	T=10	T=100
None	300K	2.61	86.4	14.43	2.93	3.28	3.27
All params	100K	2.65	79.7	17.75	3.29	3.26	2.93
New params	100K	2.66	82.5	18.91	2.34	3.14	3.25

All models were able to learn effective representations, as demonstrated by their performance on downstream tasks (AUROC). However, the finetuned models did not perform as well in generation tasks with fewer denoising steps, as evidenced by the FID scores for $T = 10$. Overall, all models perform well and are suitable for different use cases, depending on the specific trade-offs between training time, resource requirements, and generation efficiency. In Appendix A.8, we present experiments on finetuning larger-scale DMZ using pretrained models from Hugging Face.

5 Conclusion

We presented DMZ, a model inspired by the connection between diffusion autoencoders and diffusion models with a learnable forward process and designed to both learn efficiently and capture effective representations. Through targetted experimentation, we demonstrate that DMZ is capable of generating high-quality samples with fewer denoising steps, while simultaneously learning meaningful representations. Importantly, DMZ achieves these results without the need for additional loss terms, constraints on the latent variable, or auxiliary samplers. Finally, we extend DMZ to a multimodal framework and successfully apply it to an image-to-image translation task, showcasing its versatility and effectiveness. Our findings suggest that the use of additional, input-dependent priors provides a compelling and efficient alternative to traditional diffusion modelling.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Grigory Bartosh, Dmitry P. Vetrov, and Christian Andersson Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual information neural estimation, 2018.

Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 863–871. PMLR, 2018.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8780–8794, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Drew A. Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K. Lampinen, Andrew Jaegle, James L. McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. SODA: bottleneck diffusion models for representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 23115–23127. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02181.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5967–5976. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.632.

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

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. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b.

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 2015, Santiago, Chile, December 7-13, 2015*, pp. 3730–3738. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.425.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Luke Metz, Julian Ibarz, Navdeep Jaitly, and James Davidson. Discrete sequential prediction of continuous actions for deep rl, 2017.

Edan Jacob Meyer, Adam White, and Marlos C. Machado. Harnessing discrete representations for continual reinforcement learning. *Reinforcement Learning Journal*, 2:606–628, 2024.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021.

Beatrix Miranda Ginn Nielsen, Anders Christensen, Andrea Dittadi, and Ole Winther. Diffenc: Variational diffusion with a learned encoder. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26245–26265. PMLR, 2023.

Andrey Okhotin, Dmitry Molchanov, Vladimir Arkhipkin, Grigory Bartosh, Viktor Ohanesian, Aibek Alanov, and Dmitry P. Vetrov. Star-shaped denoising diffusion probabilistic models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

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)*, pp. 10619–10629, 2022.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14837–14847, 2019.

Jason Tyler Rolfe. Discrete variational autoencoders. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

Hui Shen, Jingxuan Zhang, Boning Xiong, Rui Hu, Shoufa Chen, Zhongwei Wan, Xin Wang, Yu Zhang, Zixuan Gong, Guangyin Bao, et al. Efficient diffusion models: A survey. *Transactions on Machine Learning Research (TMLR)*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Arash Vahdat, William G. Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. DVAE++: discrete variational autoencoders with overlapping transformations. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5042–5051. PMLR, 2018.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017.

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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36336–36354. PMLR, 2023a.

Ze Wang, Jiang Wang, Zicheng Liu, and Qiang Qiu. Binary latent diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 22576–22585. IEEE, 2023b. doi: 10.1109/CVPR52729.2023.02162.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 15756–15766. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01448.

Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 18892–18903. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01736.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15703–15712, 2025.

Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I-Chao Chang, and Hanwang Zhang. Exploring diffusion time-steps for unsupervised representation learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 586–595. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00068.

Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

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

Zijian Zhang, Zhou Zhao, Jun Yu, and Qi Tian. Shiftddpms: Exploring conditional diffusion models by shifting diffusion trajectories. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 3552–3560. AAAI Press, 2023b. doi: 10.1609/AAAI.V37I3.25465.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders, 2017.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Łukasz Kaiser and Samy Bengio. Discrete autoencoders for sequence models, 2018.

A Appendix

A.1 DMZ in relation to prior work

In Table 8, we show how DMZ relates to prior work on diffusion autoencoders. Algorithms 1 and 2 outline the training and sampling procedures of DMZ, detailing how the latent variable z is incorporated and highlighting differences from the original DDPM approach (Ho et al., 2020).

Table 8: Overview of model features related to training, representations, and evaluation. Diff. loss only – optimisation solely using the diffusion loss. Repr. qual. — quantitative evaluation of representation quality, e.g., via downstream task performance metrics.

	Training		**Representations**		**Evaluation**				
	E2E	Diff. loss only	Discrete	No aux sampler	HQ samples	Edits	Repr. qual.	Multi-modal	Fine-tuning
DiffAE (Preechakul et al., 2022)	✓	✓	✗	✗	✓	✓	✓	✗	✗
InfoDiff (Wang et al., 2023a)	✓	✗	✓	✗	✓	✓	✓	✗	✗
DiffuseVAE (Pandey et al., 2022)	✗	✗	✗	✗	✓	✓	✗	✗	✗
DMZ	✓	✓	✓	✓	✓	✓	✓	✓	✓

Algorithm 1 DMZ training

repeat
  Sample $x_0 \sim q(x_0)$, $t \sim \mathcal{U}(\{1, \ldots, T\})$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
  Compute noisy input $x_t \leftarrow \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}$
  Extract relaxed code z from x_0 via encoder parametrised by φ
  Take a gradient step on $\nabla_{\theta,\varphi} \left\| \epsilon - \epsilon_\theta(x_t, t, z) \right\|^2$
until convergence

Algorithm 2 DMZ sampling

Sample $\hat{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and z such that $z_i \sim \text{Bernoulli}(p = 0.5)$
for $t \leftarrow T$ to 1 **do**
  if $t > 1$ **then**
    Sample $v \sim \mathcal{N}(0, \mathbf{I})$
  else
    Set $v \leftarrow 0$
  end if
  $\hat{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\hat{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\hat{x}_t, t, z) \right) + \sigma_t v$
end for
return $\hat{x}_0$

A.2 Evidence That z Is Not Ignored

Lemma 1 (Conditional law of total variance)**.** *For random variables X, Y, and Z,*

$$\text{Var}(Y \mid X) = \mathbb{E}[\text{Var}(Y \mid X, Z) \mid X] + \text{Var}(\mathbb{E}[Y \mid X, Z] \mid X).$$

Proof. Starting from the definition of conditional variance, we have

$$\text{Var}(Y \mid X) = \mathbb{E}[Y^2 \mid X] - \mathbb{E}[Y \mid X]^2.$$

By the law of total expectation,

$$\mathbb{E}[Y^2 \mid X] = \mathbb{E}\big[\mathbb{E}[Y^2 \mid X, Z] \mid X\big],$$

and using the definition of conditional variance $\text{Var}(Y \mid X, Z)$:

$$\mathbb{E}[Y^2 \mid X, Z] = \text{Var}(Y \mid X, Z) + \mathbb{E}[Y \mid X, Z]^2.$$

Therefore,

$$\mathbb{E}[Y^2 \mid X] = \mathbb{E}[\text{Var}(Y \mid X, Z) \mid X] + \mathbb{E}\big[\mathbb{E}[Y \mid X, Z]^2 \mid X\big].$$

Substituting into the expression for $\mathrm{Var}(Y \mid X)$ yields

$$\mathrm{Var}(Y \mid X) = \mathbb{E}[\mathrm{Var}(Y \mid X, Z) \mid X] + \mathbb{E}\big[\mathbb{E}[Y \mid X, Z]^2 \mid X\big] - \mathbb{E}[Y \mid X]^2.$$

By the law of total expectation,

$$\mathbb{E}[Y \mid X] = \mathbb{E}[\mathbb{E}[Y \mid X, Z] \mid X]\,.$$

Applying the definition of conditional variance to the random variable $\mathbb{E}[Y \mid X, Z]$ conditioned on X gives

$$\mathrm{Var}(\mathbb{E}[Y \mid X, Z] \mid X) = \mathbb{E}\big[\mathbb{E}[Y \mid X, Z]^2 \mid X\big] - \mathbb{E}[Y \mid X]^2.$$

Substituting this identity into the previous expression yields

$$\mathrm{Var}(Y \mid X) = \mathbb{E}[\mathrm{Var}(Y \mid X, Z) \mid X] + \mathrm{Var}(\mathbb{E}[Y \mid X, Z] \mid X),$$

which proves the claim. □

We investigate the conditions under which the latent variable is utilized. To this end, we compare the optimal denoising estimators with and without the latent code z, corresponding to DMZ and DDPM, respectively, with minimum mean-squared errors:

$$\mathcal{L}_{\mathrm{DMZ}} = \mathbb{E}[\mathrm{Var}(\epsilon \mid x_t, z)] \qquad \mathcal{L}_{\mathrm{DDPM}} = \mathbb{E}[\mathrm{Var}(\epsilon \mid x_t)].$$

By the law of total variance applied conditionally on x_t, we have

$$\mathrm{Var}(\epsilon \mid x_t) = \mathbb{E}[\mathrm{Var}(\epsilon \mid x_t, z) \mid x_t] + \mathrm{Var}(\mathbb{E}[\epsilon \mid x_t, z] \mid x_t),$$

which yields

$$\mathcal{L}_{\mathrm{DDPM}} = \mathcal{L}_{\mathrm{DMZ}} + \mathbb{E}\left[\mathrm{Var}\left(\mathbb{E}[\epsilon \mid x_t, z] \mid x_t\right)\right].$$

The second term is non-negative and is strictly positive whenever z contains information about the clean sample that is not already present in x_t, i.e., when $I(X; Z \mid X_t) > 0$. Therefore,

$$\mathcal{L}_{\mathrm{DMZ}} \leq \mathcal{L}_{\mathrm{DDPM}}.$$

Consequently, a network that ignores z cannot achieve the minimum loss unless $Z \perp X \mid X_t$. Hence, optimization is pushed to exploit z whenever it provides additional information beyond x_t.

A.3 Effect of Regularisation Against Non-informative Prior

DMZ learns a uniform posterior that aligns with the prior, enabling sampling of z during inference. However, this alignment is not achieved for larger values of z. A reasonable solution is to apply regularisation techniques, such as a KL regularisation, to better align the distributions. Table 9 compares the performance of DMZ with and without KL regularisation.

As expected, with regularisation, the models learn to better align the posterior with the prior. However, this leads to less meaningful representations that do not improve the generative performance of the model. While good samples can be generated using the Bernoulli prior, the generative quality is weaker compared to when $|z|$ is restricted to 16 to achieve prior-posterior alignment. Moreover, regularisation negatively impacts the generative performance of the model, even for $|z| = 16$, as it interferes with the denoiser's objective.

Table 9: Performance on CIFAR-10 with and without KL regularisation

$\lvert z\rvert$	KL reg.	Acc	FID@10K	
			Bernoulli	$q_\phi(z\|x_0)$
16	✗	39.5	4.79	4.56
16	✓	35.8	5.05	5.23
64	✗	45.6	9.33	4.61
64	✓	34.9	5.40	6.05

A.4 DMZ with DDIM sampling

Our baseline DDPM is incompatible with DDIM sampling due to its use of the input perturbation technique (see Section 6.2 in (Ning et al., 2023)) within the DDPM framework[3]. To evaluate the effect of DDIM on DMZ, we trained models without the input perturbation technique. The results in Table 10 indicate that DDIM sampling and DMZ are orthogonal methods, each independently improving generative quality. Note that the DDPM without input perturbation performs worse than our baseline DDPM (see Table 1), which is why we opt to use the latter.

Table 10: Evaluation of the effect of DDIM sampling on DMZ measured via FID@10K.

Model	DDIM sampling	CIFAR-10				CelebA-64			
		10	20	50	100	10	20	50	100
DMZ	✗	15.44	8.88	5.99	5.11	28.62	13.73	6.72	4.87
	✓	12.51	8.55	6.19	6.08	20.95	12.42	6.51	4.52
DDPM	✗	22.90	11.93	6.80	5.53	73.14	29.35	11.39	8.43
	✓	13.17	8.21	5.54	5.29	20.34	14.91	10.31	9.41

A.5 Interpolation formulation and examples

Algorithm 3 describes how discrete interpolations between two latent vectors are performed by flipping the disagreeing bits one at a time in random order. For visualisations, we take latent codes equally spaced along the interpolation trajectory. Algorithm 4 details translations across the classifier's decision boundary. Examples of both are shown in Figure 8 and Figure 9.

Algorithm 3 Discrete interpolation: $z_{\text{source}} \rightarrow z_{\text{target}}$

Input: $z_{\text{source}}, z_{\text{target}} \in \{0,1\}^n$
Output: Interpolation $\{z^{(i)}\}_{i=0}^{k}$, $z^{(i)} \in \{0,1\}^n$, $z^{(0)} = z_{\text{source}}$, $z^{(k)} = z_{\text{target}}$
$\mathcal{I} = \{j \mid z_{\text{source}j} \neq z_{\text{target}j}\}$ {disagreement indices}

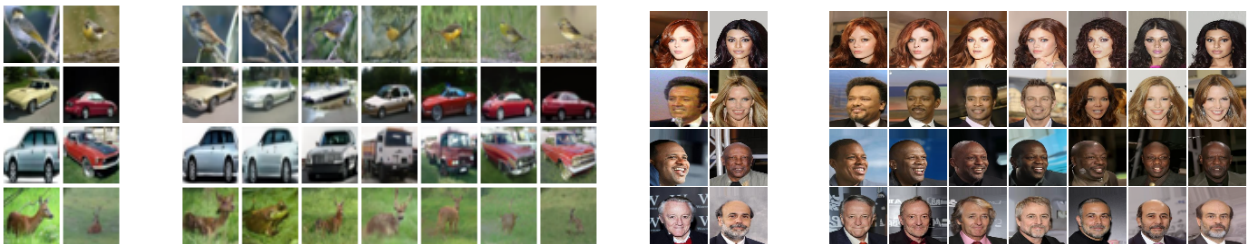
Let $k = |\mathcal{I}|$ and $j_1, \ldots, j_k$ be a random ordering of $\mathcal{I}$
$z^{(0)} \leftarrow z_{\text{source}}$
for $i \leftarrow 1$ to k **do**
  $z^{(i)} \leftarrow z^{(i-1)}$
  $z^{(i)}_{j_i} \leftarrow 1 - z^{(i)}_{j_i}$ {flip bit}
end for
return $\{z^{(i)}\}_{i=0}^{k}$

Algorithm 4 Translating z across binary decision boundary

Input: $z \in \{0,1\}^n$, classifier params $W \in \mathbb{R}^{n\times 2}$, $b \in \mathbb{R}^2$, step sizes with directions $\delta_i \in \mathbb{R}, i = 1, \ldots, k$
Output: Interpolation $\{z^{(i)}\}_{i=0}^{k}$, $z^{(i)} \in \{0,1\}^n$
Let $w_1 \leftarrow W[:,0]$, $w_2 \leftarrow W[:,1]$ {class weights}
$\mathbf{n} \leftarrow w_1 - w_2$ {normal vector to decision boundary}

$\mathbf{v} \leftarrow \frac{\mathbf{n}^\top z + b_1 - b_2}{\|\mathbf{n}\|^2} \cdot \mathbf{n}$ {translation vector}
for $i \leftarrow 1$ to k **do**
  $z^{(i)} \leftarrow z + \delta_i \mathbf{v}$ {translation of z}
  $z^{(i)} \leftarrow \mathbb{I}[z^{(i)} > 0.5]$ {optional: binarise vector}
end for
return $\{z^{(i)}\}_{i=0}^{k}$

[3] https://github.com/openai/guided-diffusion



(a) DMZ-128 for CIFAR-10 (b) DMZ-256 for CelebA-64

Figure 8: Examples of discrete interpolations between 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.

(a) Input images $x_0 \sim \mathcal{D}$

(b) $z \sim q_\varphi(z|x_0)$ with $x_T \sim \mathcal{N}(0, \mathbf{I})$

(c) $z \sim q_\varphi(z|x_0)$ with $x_t \sim q(x_t|x_0)$ and $t = 90$

Figure 9: Examples of classifier-based edits for $T = 100$ on CelebA-64 using DMZ-256. For the first image x_0, we change following CelebA attributes: glasses, male, hat; for the second: gray hair, bald, smile; the third: bangs, blond hair, hat; the fourth: bangs, male, earrings; the fifth: blond hair, male, smile.

A.6 Multimodal DMZ details

To build the multimodal framework—specifically the image-to-image model composed of DMZ modules—we train each component independently and evaluate its performance in isolation. This modular approach allows us to assess the effectiveness of each part before assembling the full model, ensuring that all components function reliably. Below, we describe this process for the multimodal DMZ trained for Edges2Handbags sketch-to-photo task.

DMZ modules We use two instances of DMZ-512: one trained on Edges-64 and the other on Handbags-64. In the sketch-to-photo task, only the model trained on photos is used to generate images, while the model trained on sketches is used to encode their representations. The mean squared error (MSE), defined as $\| x_{\text{photo}} - \widehat{x}_{\text{photo}} \|$, where $\widehat{x}_{\text{photo}} \sim p_\theta(x_{\text{photo}}|z_{\text{photo}})$ and $z_{\text{photo}} \sim q_\varphi(z|x_{\text{photo}})$, serves as an upper bound of the MSE for the sketch-to-photo generation task. We monitor this metric during training and stop once it no longer improves. Additionally, the latent dimensionality $|z| = 512$ was selected based on that MSE performance. Figure 10 shows the reconstruction error over the course of training.

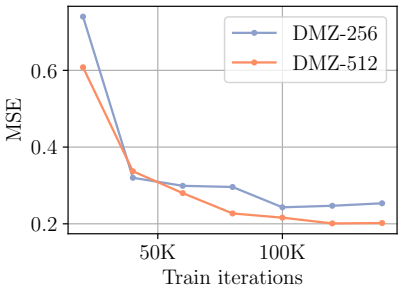


Figure 10: Reconstruction error of DMZ models trained on Handbags-64 measured on 10K images, an upper bound for Edges2Handbags task.

Mapping γ We train an MLP to learn a mapping $\gamma : Z_{\text{sketch}} \rightarrow Z_{\text{photo}}$ using latent codes from the DMZ models. To determine the optimal architecture, we experiment with different numbers of layers L in the MLP and evaluate MSE $\| x_{\text{photo}} - \widehat{x}_{\text{photo}} \|$, where $\widehat{x}_{\text{photo}} \sim p_\theta(x_{\text{photo}} \mid \gamma(z_{\text{sketch}}))$ and $z_{\text{sketch}} = q_\varphi(z \mid x_{\text{sketch}})$. The resulting MSEs for $L = 1, 2, 4, 6, 8$ are 0.26, 0.24, 0.23, 0.22, and 0.23, respectively, leading us to select $L = 6$ as the optimal depth. However, note that a simpler mapping would provide greater interpretability for the framework.

Additional capabilities of DMZ image-to-image framework With our DMZ framework, we can perform reverse image-to-image mapping—generating sketches from photos—as well as unconditional generation of both photos and sketches. Examples are shown in Figure 11. Note that we use PixelSNAIL for unconditional generation, as the latent size $|z| = 512$ was chosen to optimise reconstruction loss rather than efficient sampling of z.

(a) Photo-to-sketch translations (b) Unconditional generation

Figure 11: Qualitative results showing additional capabilities of the DMZ image-to-image framework.

A.7 Reconstruction Performance of Discrete vs Continuous Representations

A continuous latent variable z generally carries more information about the target x_0 than a discrete latent of the same size, enabling better reconstruction. However, this advantage holds primarily when z is the only available information about the target—specifically, at the final timestep T. For intermediate noisy inputs x_t with $t < T$, Figure 12 shows that the reconstruction error becomes comparable between models with discrete and continuous latents.

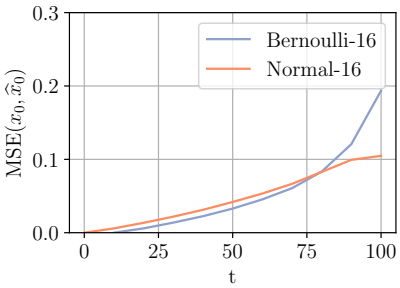


Figure 12: Reconstruction performance comparison between DMZ (Bernoulli-16) and continuous latent z (Normal-16). Mean Squared Error (MSE) is reported on CIFAR-10 for predicting x_0 from x_t over t denoising steps, with $T = 100$.

The core idea behind DMZ is to learn latent representations that provide the denoiser with additional, useful information to improve denoising efficiency, while keeping the latent space simple enough to allow easy sampling during inference. Importantly, in DMZ, the latent z is not intended to be sufficient for full image reconstruction on its own. Rather, it serves as a complementary guide for the denoiser. For downstream tasks requiring higher reconstruction fidelity, such as targeted edits or interpolations, one can utilise intermediate states x_t. Our main experiments assume $t = T$ to isolate the effect of z alone. Figure 9 illustrates examples of edits that leverage state x_t to better preserve image identity.

A.8 Finetuning HuggingFace models

We have demonstrated that unconditional DDPMs can be effectively finetuned into the DMZ framework, enabling representation learning, conditional generation, and other capabilities. Here, we finetune the publicly available DDPM model trained on CelebA-HQ (256×256), available via HuggingFace[4], into DMZ. We train our models for 40K training iterations by finetuning all parameters. Quantitative results are presented in Table 11.

Table 11: Evaluation of DDPM trained on CelebA-HQ and DMZ finetuned from it.

Model	NLL (BPD)	AUROC	FID@10K				IS@10K
			T=10	T=20	T=50	T=100	T=100
DDPM	6.25	—	71.43	53.55	36.86	29.81	2.77
DMZ-64	3.01	74.06±0.14	39.91	28.16	19.60	15.15	3.06
DMZ-256	3.00	88.53±0.05	49.53	42.31	33.25	27.54	2.59

A.9 DMZ with Diffusion Transformers

To demonstrate the versatility of the DMZ framework, we apply it to the attention-based DiT architecture (Peebles & Xie, 2022). We adapt the model by replacing self-attention layers in every second DiT block with cross-attention layers that attend to a latent z.

We train DiT-B/4 models on CIFAR-10 for 200K iterations without specific hyperparameter tuning[5]. As shown in Table 12, DMZ significantly improves generative quality at lower inference steps compared to the baseline DiT, while maintaining an identical NLL. Furthermore, the learned representations achieve an unsupervised classification accuracy, surpassing the U-Net-based results reported in Table 2.

[4]https://huggingface.co/google/ddpm-ema-celebahq-256/
[5]https://github.com/facebookresearch/DiT

Table 12: Comparison of FID scores on CIFAR-10 between DiT and DiT adapted with DMZ.

Model	NLL (BPD)	Acc	FID@50K				IS@50K
			$T = 10$	$T = 20$	$T = 50$	$T = 100$	$T = 100$
DiT-B/4	3.30	–	32.14	15.56	10.79	10.15	8.52
DMZ-16	3.30	51.52	17.76	12.28	10.81	10.91	8.32

A.10 DMZ with ImageNet

Additionally, we train DMZ and baseline DDPM models on ImageNet 32×32 to evaluate performance on a larger-scale dataset (1.28M images, 1K classes). We use the same training setup as CIFAR-10 due to the identical resolution, without further hyperparameter tuning. Results are reported in Table 13.

Table 13: Evaluation of DMZ trained on ImageNet (32×32).

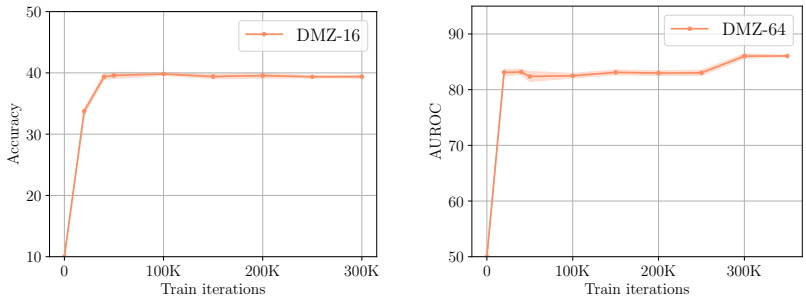
Model	train iter.	NLL (BPD)	Acc	FID@50K				IS@50K
				$T = 10$	$T = 20$	$T = 50$	$T = 100$	$T = 100$
DDPM	850K	4.15	–	37.04	13.14	6.56	5.75	12.14
DMZ-16	450K	4.13	9.34	19.30	8.54	6.22	6.17	12.13
DMZ-32	450K	4.13	11.25	17.67	10.22	8.16	7.82	11.25
DMZ-64	450K	4.12	8.48	16.63	12.67	11.73	12.02	10.32

A.11 Additional scores and samples

Additional scores are reported in Table 14 (FID@10K) Table 15 (Inception Scores) Table 16 (reconstruction metrics), Figure 13 (accuracy over training iterations), and additional examples are provided in Figures 14 to 16.

Table 14: FID scores for DMZ models evaluated with 10K samples. Reported values are the average $\pm$ standard deviation over 5 models.

Model	Dataset	T=10	T=20	T=50	T=100
DMZ-16	CIFAR-10	11.53±0.90	6.76±0.69	4.96±0.36	4.67±0.24
DMZ-64	CelebA-64	15.70±0.50	8.91±0.38	5.18±0.27	3.84±0.48

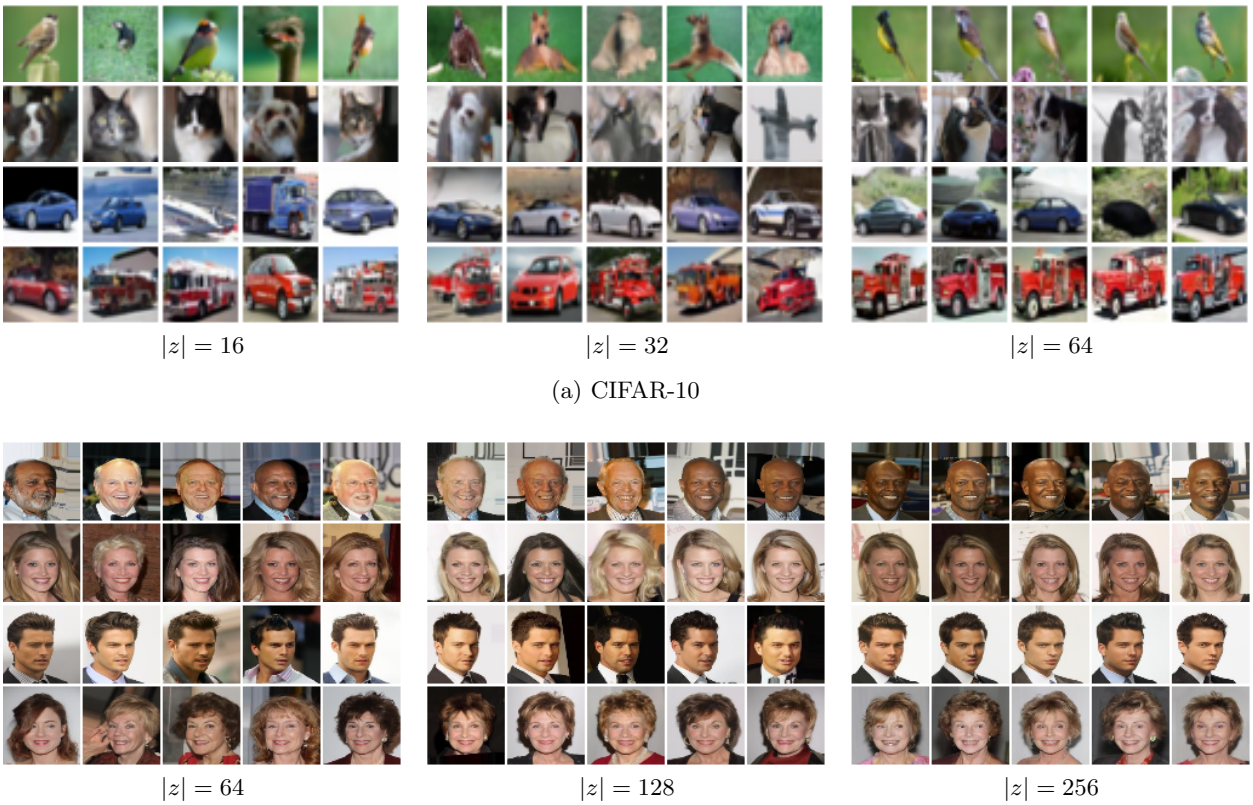


(a) CIFAR-10 (b) CelebA-64

Figure 13: Evolution of representation quality during training. High-quality representations emerge early and remain stable throughout.

A.12 Reproducibility Details

We adopt the hyperparameter settings from Ning et al. (2023), which are based on the configurations by Dhariwal & Nichol (2021). The specific values are listed in Table 17. All models are trained using the AdamW



$|z| = 16$ $|z| = 32$ $|z| = 64$

(a) CIFAR-10

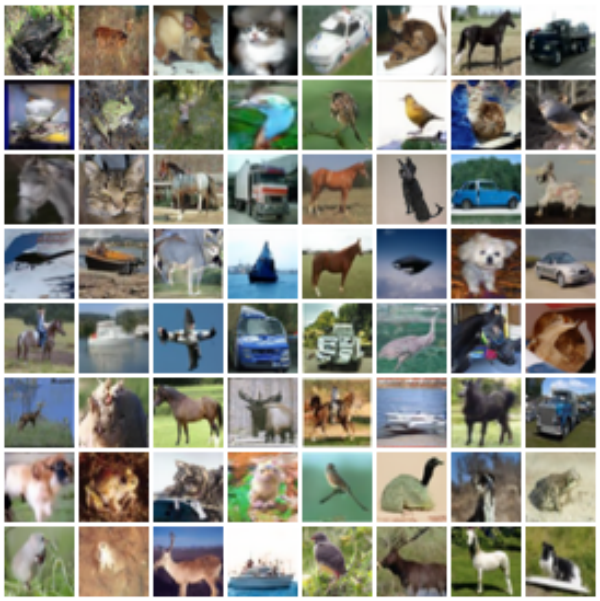
$|z| = 64$ $|z| = 128$ $|z| = 256$

(b) CelebA-64

Figure 14: Comparison of representations learned by DMZ on CIFAR-10 and CelebA-64 for varying latent sizes $|z|$. Images are generated from $z \sim q_\phi(z \mid x_0)$, $x_0 \sim \mathcal{D}$ and five different $x_T \sim \mathcal{N}(0, \mathbf{I})$.

(a) DMZ-16 for CIFAR-10 (b) DMZ-64 for CelebA-64

Figure 15: Images generated with varying numbers of denoising steps T. Each column shows samples generated from a fixed latent code $z \sim$ Bernoulli. Rows correspond to $T = 1000, 500, 200, 100, 50, 20, 10, 5$ steps, from top to bottom.



(a) DMZ-16 for CIFAR-10

(b) DMZ-64 for CelebA-64

(c) DMZ-64 for CelebA-HQ

Figure 16: Generated sample images produced using $T = 100$ diffusion steps.

Table 15: Inception Scores (IS) measured for 50K generated samples. Note that DiffuseVAE is the only baseline that evaluates IS.

Model	T	CIFAR-10	CelebA-64
DMZ	10	9.10±0.08	3.33±0.12
	20	9.39±0.08	3.36±0.10
	50	9.59±0.14	3.32±0.08
	100	9.64±0.07	3.33±0.10
DDPM (Ning et al., 2023) (reproduced)	10	8.36	2.82
	20	9.19	3.94
	50	9.38	3.01
	100	9.42	3.10
DiffuseVAE (Pandey et al., 2022)	1000	9.60	–

Table 16: Reconstruction metrics achieved by DMZ. Comparison against the test set.

Dataset	$\|z\|$	SSIM	LPIPS	MSE	PSNR	IS	FID
CIFAR-10	16	0.090	0.624	0.378	10.277	9.705	4.618
	32	0.119	0.596	0.347	10.647	9.639	4.821
	64	0.225	0.556	0.248	12.150	9.529	4.798
	128	0.374	0.510	0.139	14.712	9.395	4.763
CelebA-64	32	0.106	0.541	0.534	8.742	3.137	3.864
	64	0.145	0.493	0.483	9.183	3.223	3.170
	128	0.224	0.447	0.405	9.953	3.294	3.593
	256	0.324	0.403	0.298	11.298	3.264	3.092

optimiser Loshchilov & Hutter (2019) with 16-bit mixed precision training with loss scaling (Micikevicius et al., 2018; Dhariwal & Nichol, 2021), while keeping the model weights, EMA, and optimiser states in 32-bit precision. An EMA decay rate of 0.9999 is used in all experiments, following the setup from Ning et al. (2023).

The encoder used to extract codes z from input images consists of repeated blocks of a convolutional layer, batch normalisation, and LeakyReLU activation, followed by a final projection layer. We use 4 blocks for 32×32 images, 5 blocks for 64×64, and 7 blocks for 256×256.

We use PyTorch Paszke et al. (2019), and train all models with Python 3.10 and PyTorch version 2.6. For CIFAR-10, we use a single NVIDIA A40 GPU and train for approximately 2 days. For CelebA-64, we use two A40 GPUs and train for about 10 days. Models trained on Edges2Handbags—handled separately as Edges and Handbags—are also trained using two A40 GPUs, with a training time of around 3 days. Finetuning of CelebA-HQ DDPM (Ho et al., 2020) takes slightly less than 2 days on 4 A40 GPUs. For T=100, sampling a batch of 64 images using a single A40 GPU takes 12s, 264.5s, 79.3s, 79.3s, and 222.5s, for CIFAR-10, CelebA, Edges, Handbags, and CelebA-HQ, respectively.

Our implementation and instructions for reproducing the experiments are available at https://anonymous.4open.science/r/dmz-9D08, and will be made public upon publication.

Table 17: Hyperparameter values based on Ning et al. (2023) for all datasets except CelebA-HQ, where we follow the configuration from Ho et al. (2020) and perform finetuning only.

	CIFAR-10 32×32	CelebA 64×64	Edges 64×64	Handbags 64×64	CelebA-HQ* 256×256
Size of z	16/32/64	64/128/256	512	512	64/256
Diffusion steps	1,000	1,000	1,000	1,000	1,000
Noise schedule	cosine	cosine	cosine	cosine	linear
UNet size	69M	409M	333M	333M	142M
Encoder size	0.5M	1.8M	3.7M	3.7M	29M
Channels	128	192	192	192	128
Residual blocks	3	3	3	3	2
Channels multiple	1, 2, 2, 2	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 1, 2, 2, 4, 4
Heads channels	32	64	64	64	512
Attention resolution	16, 8	32, 16, 8	32, 16, 8	32, 16, 8	16
Cross attention resolution	16, 8	32, 16, 8	16	16	16
Mid-block cross attention	True	True	True	True	True
BigGAN up/downsample	True	True	True	True	True
Dropout	0.3	0.1	0.1	0.1	0.1
Batch size	128	256	256	256	256
Training iterations	250K	300K	120K	120K	40K
Training images	50K	163K	139K	139K	24K
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4
Learned sigma	True	True	True	True	False
Noise schedule	cosine	cosine	cosine	cosine	linear
Input perturbation	0.15	0.1	0.1	0.1	0.1