HYBRID DIFFUSION: LATENT-TO-PIXEL DIFFUSION FOR LIMITED DATA GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel diffusion models have achieved remarkable success in synthesizing fine textures and local structures by denoising score matching. Despite this success, learning accurate score functions in such a high-dimensional space requires substantial data, making it difficult to reliably capture global semantic and structural information under limited data regimes. Latent diffusion addresses this challenge by removing pixel-level redundancy and performing diffusion in a compressed representation that retains salient semantic and structural information. However, because the autoencoder is typically frozen and lossy, this compression can limit the fidelity of fine-grained details. In this paper, we introduce Hybrid Diffusion, a hybrid latent-to-pixel diffusion framework that effectively combines the advantages of both pixel and latent diffusion models for limited data generation. Specifically, Hybrid Diffusion consists of both latent and pixel stages, and performs most of the early denoising in the latent stage before switching to the pixel stage at a chosen transition point, thereby substantially narrowing the pixel-space learning problem while addressing the representation bottleneck of latent space. Importantly, we make the pixel stage explicitly aware of latent prediction errors by using decoded clean predictions from randomly forward-corrupted training latents as training conditions, allowing the pixel stage to address these errors alongside autoencoder reconstruction artifacts. Experiments across multiple datasets demonstrate consistent improvements over both pixel and latent diffusion baselines.

1 INTRODUCTION

Diffusion models have achieved remarkable sample quality and diversity across unconditional image synthesis (Vahdat et al., 2021; Song et al., 2023; 2020; Nichol & Dhariwal, 2021) and text-to-image generation (Podell et al., 2023; Rombach et al., 2022; Nichol et al., 2021; Saharia et al., 2022). Diffusion models typically rely on large-scale datasets to learn accurate score functions through denoising score matching, making training challenging in limited data regimes (Dhariwal & Nichol, 2021; Ho et al., 2020; Karras et al., 2022; Hyvärinen & Dayan, 2005). Collecting and cleaning data at this scale is often costly and time-consuming, raises substantial privacy concerns, and can be infeasible in specialized domains where examples are intrinsically scarce or difficult to collect (Zhang et al., 2025; Karras et al., 2020a; Carlini et al., 2023). Developing diffusion models that can learn reliably from limited data is therefore important for extending their applicability beyond data-rich settings (Moon et al., 2022; Zhu et al., 2022; Giannone et al., 2022).

In limited data settings (Wang et al., 2020), the choice of modeling space is particularly important, as it affects both the difficulty of score estimation and the image information retained for generation. Pixel diffusion can synthesize fine textures and local structures, but estimating scores in a high-dimensional image space from few examples can make it difficult to learn global structure reliably. In contrast, latent diffusion alleviates this difficulty by removing pixel-level redundancy and modeling compact representations that emphasize semantic and structural information. This advantage comes at the cost of a representation bottleneck: the lossy encoder may discard local details, while the frozen decoder restricts the range of variations that can be reconstructed. As a result, generative fidelity can be limited (Zheng et al., 2026). Directly combining pixel diffusion and latent diffusion is fundamentally limited by the mismatch between the two modeling spaces, noise processes, and error distributions. Latent diffusion produces decoded samples that contain both errors from imperfect latent prediction and artifacts introduced by the autoencoder, whereas pixel diffusion is not

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) EDM + DA (Pixel) **(b) LD-Diffusion (Latent)** **(c) Hybrid Diffusion**

Figure 1: The comparison of generated images on the 100-shot Obama dataset: (a) EDM + DA (Pixel), (b) LD-Diffusion (Latent), and (c) Hybrid Diffusion. Representative fine-grained regions are highlighted with red boxes for visual comparison.

generally trained on this coupled error distribution. Consequently, the pixel-space model cannot reliably correct errors inherited from latent generation without being exposed to a broad range of high-dimensional pixel-space variations. This would reintroduce much of the distribution-estimation burden that latent compression is intended to avoid, preventing a straightforward combination from simultaneously preserving latent diffusion's ability to model global structure and pixel diffusion's ability to recover fine details.

To address this challenge, we introduce Hybrid Diffusion, a hybrid latent-to-pixel diffusion framework that effectively combines the advantages of latent and pixel diffusion models for limited data generation. Specifically, Hybrid Diffusion performs most early denoising in latent space before switching to a conditional pixel-space denoiser at a chosen transition point. At the switch, the latent model's predicted clean latent is decoded into an image-space estimate that conditions the remaining pixel-space refinement steps. Because this decoded estimate already contains the main image structure, the pixel denoiser learns a conditional refinement distribution around it rather than the full pixel-space distribution from noise, substantially narrowing the pixel-space learning problem while addressing the representation bottleneck of latent space. Importantly, we use decoded clean predictions from randomly forward-corrupted training latents as training conditions. This makes pixel-space refinement explicitly aware of latent prediction errors, enabling the denoiser to address these errors alongside autoencoder reconstruction artifacts. As shown in Figure 1, Hybrid Diffusion can achieve better visual results compared with both the representative pixel diffusion model (EDM + DA Karras et al. (2022)) and latent diffusion model (LD-Diffusion Zhang et al. (2025)) under limited data. Our main contributions are summarized as follows:

(1) We propose Hybrid Diffusion, a hybrid latent-to-pixel diffusion framework for limited data generation that performs most early denoising in latent space to capture global semantics and structure, then switches to conditional pixel-space refinement for fine details. It offers a practical way to combine diffusion spaces and informs future research.

(2) We design an error-aware conditional refinement strategy that enables the latent-to-pixel transition to compensate for errors introduced by latent prediction and autoencoder reconstruction. By conditioning the pixel denoiser on decoded clean estimates from randomly forward-corrupted latent states, the model learns a conditional refinement distribution around an approximate decoded estimate rather than the full pixel-space distribution from noise, thereby narrowing the pixel-space learning problem in limited data regimes.

(3) Extensive experiments on multiple datasets show that Hybrid Diffusion consistently outperforms both pixel and latent baselines across several commonly-used metrics, demonstrating the superiority and effectiveness of proposed Hybrid Diffusion.

2 Related Work

Diffusion Models. Diffusion models have become a dominant family of generative models for high-quality image synthesis (Ho et al., 2022). Denoising Diffusion Probabilistic Models (DDPMs)

formulate generation as the reverse process of gradual noising (Ho et al., 2020), while score-based generative models learn scores of perturbed data distributions for sampling (Song & Ermon, 2019; Song et al., 2021). Advanced works improve diffusion models through better architectures, training objectives, and samplers (Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Peebles & Xie, 2023; Karras et al., 2024).

Diffusion Models with Limited Data. Recent work improves training diffusion models with limited data from scratch through data augmentation, regularization, patch-based learning, and objective modifications, enabling better generalization on small datasets without large pretrained models (Wang et al., 2023; 2026; Yang et al., 2024; Fu et al., 2024; Sinha et al., 2021). Another related direction reduces the complexity of the denoising problem by training diffusion models in latent or compressed spaces (Vahdat et al., 2021; Rombach et al., 2022; Esser et al., 2021). The role of compression in limited-data diffusion is further examined by LD-Diffusion, which analyzes the problem from the perspective of hypothesis-space complexity and shows that compression can improve sample efficiency (Zhang et al., 2025).

Pixel and Latent Diffusions. Diffusion models can operate in either pixel space or latent space. Pixel diffusion models, such as DDPM and ADM, directly denoise RGB images and can preserve fine textures and local structures (Ho et al., 2020; Dhariwal & Nichol, 2021). However, score estimation in the high-dimensional image space requires substantial data, making it difficult to learn global semantic and structural information reliably in limited-data settings (Karras et al., 2022; Zhang et al., 2025). On the contrary, Latent diffusion models, such as LSGM and LDM, instead perform denoising in compact representations produced by pretrained autoencoders (Vahdat et al., 2021; Rombach et al., 2022). By removing pixel-level redundancy, they reduce the complexity of score estimation and improve statistical efficiency (Chen et al., 2023; Zhang et al., 2025; Boffi et al., 2024; Wang et al., 2024). This compression, however, introduces a representation bottleneck: the encoder may discard local details, while the frozen decoder can limit reconstruction fidelity (Rombach et al., 2022; Zheng et al., 2026). These complementary strengths and limitations motivate Hybrid Diffusion, which performs most denoising in latent space to capture global structure and uses pixel-space refinement to recover fine details.

3 Methodology

3.1 Analysis of Denoising Score Matching with Limited Data

Building on the finite-sample analysis of LD-Diffusion (Zhang et al., 2025), we use the Empirical Risk Minimization (ERM) framework (Shalev-Shwartz & Ben-David, 2014; Bartlett & Mendelson, 2002; Mohri et al., 2018) to analyze pixel and latent diffusion models under limited data. For both pixel and latent space models, the population risk decomposes into irreducible, approximation, and estimation terms, with only the estimation term depending explicitly on the number of training samples when the data distribution and hypothesis class are fixed. Under standard boundedness assumptions, the estimation term is controlled by the effective complexity of the corresponding denoising hypothesis class, as formalized in Appendix A.1.

Pixel diffusion. Let $\mathcal{H}_{\mathrm{pix}}$ denote the hypothesis class of a pixel diffusion, which directly models noisy images and produces samples in the full image space. Let ℓ_{pix} denote the per-sample pixel-space denoising loss, and let $\ell_{\mathrm{pix}} \circ \mathcal{H}_{\mathrm{pix}}$ denote the loss class induced by $\mathcal{H}_{\mathrm{pix}}$. Therefore, for N i.i.d. training samples, its estimation error can be bounded as

$$\mathcal{E}_{\mathrm{est}}^{\mathrm{pix}} \lesssim \mathfrak{R}_N(\ell_{\mathrm{pix}} \circ \mathcal{H}_{\mathrm{pix}}) \lesssim \sqrt{\frac{\mathcal{C}(\mathcal{H}_{\mathrm{pix}})}{N}}, \tag{1}$$

where $\mathcal{E}_{\mathrm{est}}^{\mathrm{pix}}$ denotes the finite-sample estimation error, $\mathfrak{R}_N(\ell_{\mathrm{pix}} \circ \mathcal{H}_{\mathrm{pix}})$ denotes the Rademacher complexity of the induced loss class on N samples, $\mathcal{C}(\mathcal{H}_{\mathrm{pix}})$ denotes the effective complexity of the pixel-space denoising problem, and $a \lesssim b$ means that $a \leq cb$ for a universal constant c independent of N. The main advantage of pixel diffusion is that it is not restricted to the range of a decoder and can directly model fine textures, local structures, and image-space variations. However, the high dimensionality of pixel space generally leads to a larger hypothesis class complexity. With limited

data, the resulting estimation error can dominate, making it difficult to learn global semantic and structural information reliably.

Latent diffusion. Latent diffusion first maps a clean image x_0 to a compact representation,

$$z_0 = s \cdot \mathrm{Enc}(x_0), \qquad g(z) = \mathrm{Dec}(z/s), \tag{2}$$

where $s > 0$ is a fixed latent scaling factor. Here, Enc and Dec denote the pretrained encoder and decoder, respectively. Their parameters are kept fixed throughout training. Let $\mathcal{H}_{\mathrm{lat}}$ denote the latent-space denoising hypothesis class, and let ℓ_{lat} denote the per-sample latent-space denoising loss for the chosen prediction target, with $\ell_{\mathrm{lat}} \circ \mathcal{H}_{\mathrm{lat}}$ denoting the corresponding induced loss class. Compression can reduce the effective complexity of the denoising problem, leading to the analogous bound

$$\mathcal{E}_{\mathrm{est}}^{\mathrm{lat}} \lesssim \mathfrak{R}_N(\ell_{\mathrm{lat}} \circ \mathcal{H}_{\mathrm{lat}}) \lesssim \sqrt{\frac{\mathcal{C}(\mathcal{H}_{\mathrm{lat}})}{N}}, \tag{3}$$

where $\mathcal{C}(\mathcal{H}_{\mathrm{lat}})$ is often much smaller than $\mathcal{C}(\mathcal{H}_{\mathrm{pix}})$. This gives latent diffusion an advantage in limited data regimes by reducing the estimation burden and allowing the model to learn global semantic and structural information more efficiently. However, this advantage comes with an image representation error. For any estimate $\hat{z}_0$ of the clean latent z_0 associated with x_0, assume that g is L_g-Lipschitz over a region containing z_0 and $\hat{z}_0$. Then

$$\|x_0 - g(\hat{z}_0)\|_2 \le \underbrace{\|x_0 - g(z_0)\|_2}_{e_{\mathrm{rep}}} + \underbrace{\|g(z_0) - g(\hat{z}_0)\|_2}_{e_{\mathrm{pred}}}, \tag{4}$$

where

$$e_{\mathrm{pred}} \le L_g \|z_0 - \hat{z}_0\|_2.$$

The expectations below are taken over the data distribution and any randomness in latent prediction or encoder sampling, when applicable. Consequently,

$$\mathbb{E}\big[\|x_0 - g(\hat{z}_0)\|_2^2\big] \le 2 \underbrace{\mathbb{E}[e_{\mathrm{rep}}^2]}_{\text{representation error}} + 2L_g^2 \underbrace{\mathbb{E}\big[\|z_0 - \hat{z}_0\|_2^2\big]}_{\text{latent prediction risk}}. \tag{5}$$

The first term is determined by the frozen autoencoder and cannot be reduced by improving the latent denoiser. Moreover, latent-only outputs are confined to $\mathrm{range}(g)$, which may limit the recovery of fine-grained variations that the frozen decoder cannot represent. As discussed in Appendix A.2, the decoded condition $g(\hat{z}_0)$ can contain both reconstruction and latent-prediction residuals.

To sum up, pixel diffusion therefore provides greater image-space expressiveness but faces a larger estimation error, whereas latent diffusion improves sample efficiency but introduces a representation bottleneck. This motivates a hybrid formulation in which the latent stage captures the initial global structure and a pixel stage performs restricted-noise refinement around the decoded latent prediction.

3.2 Hybrid Latent-to-Pixel Diffusion

To combine the advantages of pixel and latent diffusion models, we propose Hybrid Diffusion, which combines an initial latent generative stage with conditional pixel refinement over a restricted noise interval. The overview of Hybrid Diffusion is shown in Figure 2, where the *latent stage* models image structure and semantic content in a compact representation, while the *pixel stage* models the fine-grained details. During training, a clean latent estimate is obtained from a forward-corrupted latent at a sampled latent noise level and decoded into an image space condition. During inference, let σ denote the current noise level, and let σ_{switch} denote the predefined noise threshold at which the sampler transitions to the pixel stage. The conditioning signal remains fixed during this transition. The pixel stage is initialized from the current noisy state at $\sigma = \sigma_{\mathrm{switch}}$.

The pixel stage performs conditional refinement over the remaining restricted noise interval, rather than full-range generation from pure pixel-space noise. The condition used by the pixel stage is generated from the latent stage's own predictions rather than from an ideal autoencoder reconstruction. For each training example, we perturb the latent code at a sampled noise level, predict the clean latent, and decode it into an image-space condition. This condition contains both autoencoder reconstruction artifacts and latent-prediction errors, exposing the pixel stage to the imperfections it

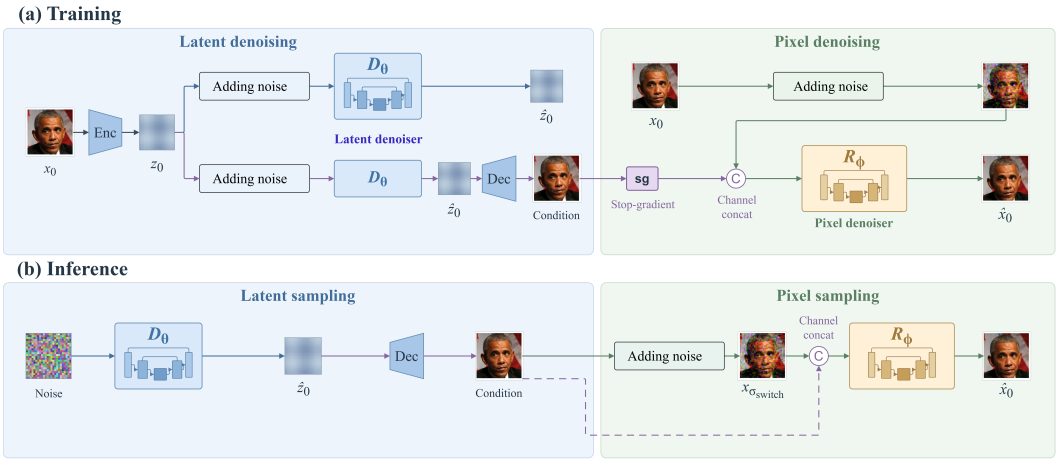


Figure 2: Overview of Hybrid Diffusion. (a) The training process of Hybrid Diffusion (b) The sampling process of Hybrid Diffusion.

encounters during sampling. The pixel denoiser is trained to recover the clean image from an independently noisy pixel observation while using the decoded prediction as an additional condition. Because the final output is produced directly in pixel space, it is not constrained to $\mathrm{range}(g)$, the set of images that the autoencoder decoder g can produce. This allows Hybrid Diffusion to recover image-space details that direct autoencoder decoding may miss.

3.2.1 Latent Denoising Stage

Following LD-Diffusion (Zhang et al., 2025), we retain the frozen pretrained encoder–decoder of Eq. (2) and perform the primary generative process in its latent space. Let $x_0 \in \mathbb{R}^{3\times H\times W}$ denote a training image with latent code z_0. For a latent model, a noisy latent is constructed as

$$z_\sigma = z_0 + \sigma\epsilon_z, \qquad \epsilon_z \sim \mathcal{N}(0, I), \tag{6}$$

and the latent denoiser D_θ predicts the corresponding clean latent. We optimize

$$\mathcal{L}_{\mathrm{lat}} = \mathbb{E}_{\sigma,\epsilon_z,x_0}\left[w_{\mathrm{lat}}(\sigma)\,\|D_\theta(z_\sigma,\sigma) - z_0\|_2^2\right], \tag{7}$$

where $w_{\mathrm{lat}}(\sigma)$ follows the noise-dependent weighting used by the latent baseline. The displayed expectation is the weighted denoising component used by the latent stage.

3.2.2 Model-Produced Cross-Stage Conditioning

We condition the pixel denoiser on decoded predictions from the latent denoiser. Specifically, during pixel stage training, we sample a positive latent noise level σ_c from the noise distribution and corrupt the clean latent as

$$z_c = z_0 + \sigma_c\epsilon_c, \qquad \epsilon_c \sim \mathcal{N}(0, I), \tag{8}$$

then obtain a predicted clean latent $\hat{z}_{0,c} = D_\theta(z_c, \sigma_c)$ and decode it to form

$$c_c = g(\hat{z}_{0,c}). \tag{9}$$

Unlike a reconstruction-only condition $g(z_0)$, the residual $x_0 - c_c$ decomposes into reconstruction and prediction components, so the pixel denoiser is trained on model-produced decoded estimates rather than idealized reconstructions. This uses forward-corrupted surrogates rather than sampled latent trajectories; the corresponding condition distributions need not be identical (Appendix A.3).

We apply a stop-gradient before the condition enters the pixel stage,

$$\bar{c}_c = \mathrm{sg}(c_c), \tag{10}$$

so that optimization of the pixel objective does not modify the latent stage. A further discussion of this design is provided in Appendix A.5.

3.2.3 CONDITIONAL PIXEL SPACE REFINEMENT

Given the model-produced condition $\bar{c}_c$, the pixel denoiser performs conditional denoising over a restricted noise interval rather than modeling the full image distribution from high-noise initialization. For a pixel noise level σ_p, we construct

$$x_{\sigma_p} = x_0 + \sigma_p \epsilon_p, \qquad \epsilon_p \sim \mathcal{N}(0, I), \tag{11}$$

where $0 < \sigma_{p,\min} < \sigma_{p,\max}$ and $\log \sigma_p \sim \mathcal{U}(\log \sigma_{p,\min}, \log \sigma_{p,\max})$. The pixel noise is sampled independently of the clean image and the variables used to construct $\bar{c}_c$. The conditional pixel denoiser R_ϕ predicts the clean image as $\hat{x}_0 = R_\phi(x_{\sigma_p}, \bar{c}_c, \sigma_p)$ and is optimized using

$$\mathcal{L}_{\text{pix}} = \mathbb{E}_{\sigma_p, \sigma_c, \epsilon_c, \epsilon_p, x_0} \left[\left\| R_\phi(x_{\sigma_p}, \bar{c}_c, \sigma_p) - x_0 \right\|_2^2 \right], \tag{12}$$

which is an unweighted pixel-space mean-squared error over the configured low-noise range. The expectation is taken over training images and the sampled variables $(\sigma_c, \epsilon_c, \sigma_p, \epsilon_p)$. During training, the decoded condition $\bar{c}_c$ is provided separately, while x_{σ_p} is obtained by corrupting the clean image x_0. Accordingly, R_ϕ acts as a conditional pixel-space refinement module over $[\sigma_{p,\min}, \sigma_{p,\max}]$: the latent stage establishes the global image structure, whereas R_ϕ focuses on correcting image-space errors caused by imperfect $\bar{c}_c$. The associated statistical trade-off is discussed in Appendix A.5.

Coupled training. Because c_c is produced by the current latent denoiser, its distribution is non-stationary while D_θ is still changing rapidly. We therefore warm up the latent stage for K_w before activating the pixel objective, after which the two are optimized concurrently:

$$\mathcal{L} = \mathcal{L}_{\text{lat}} + \mathbb{I}[k \geq K_w] \cdot \lambda_p \mathcal{L}_{\text{pix}}, \tag{13}$$

where k denotes the current training iteration. Because the conditioning signal is detached as specified in Eq. (10), gradient propagation through the conditioning path is blocked. Assuming that θ and ϕ parameterize the latent and pixel stages, respectively, and that no alternative differentiable paths exist, the cross-branch derivatives vanish:

$$\nabla_\theta \mathcal{L}_{\text{pix}} = \mathbf{0}, \qquad \nabla_\phi \mathcal{L}_{\text{lat}} = \mathbf{0}. \tag{14}$$

The joint loss in Eq. (13) is optimized using a single Adam optimizer in our implementation, where λ_p denotes the pixel-loss coefficient. The branches remain coupled in the forward pass: each pixel update uses a condition generated by the current latent model, allowing the pixel stages to track the latent stage as it improves.

3.2.4 LATENT-TO-PIXEL TRANSITION AT INFERENCE

During sampling, a shared numerical noise grid is used for both branches. We set the switch coordinate to $t_{\text{cur}} = \sigma_{\text{switch}}$. At this level, we let z_s denote the current latent state and form the clean-latent prediction $\hat{z}_{0,s} = D_\theta(z_s, t_{\text{cur}})$, then decode it into the sampling-time condition

$$c_s = g(\hat{z}_{0,s}). \tag{15}$$

Rather than using c_s directly as the initial pixel state, we initialize refinement with the current shared noise level:

$$x_{t_{\text{cur}}} = c_s + t_{\text{cur}} \epsilon_x, \qquad \epsilon_x \sim \mathcal{N}(0, I), \tag{16}$$

where t_{cur} is the shared sampler noise level at the switch. The condition c_s then remains fixed throughout the remaining pixel trajectory, which follows the shared EDM schedule. Because training corrupts x_0 whereas sampling perturbs c_s, this initialization is an approximate conditional-refinement strategy rather than an exact match to the training conditional marginal (Appendix A.4). All positive noise levels at which the pixel denoiser is evaluated after the switch should lie in $[\sigma_{p,\min}, \sigma_{p,\max}]$; the zero-noise endpoint is not a positive-noise training level.

4 EXPERIMENTS

We select FFHQ (Karras et al., 2019; 2020b), low-shot (Zhao et al., 2020), BRECAHAD datasets (Aksac et al., 2019) for the experiments. For fair comparisons, we follow the official open-source code of EDM[1] for preprocessing and resizing FFHQ to 256×256 and BRECAHAD to

[1] https://github.com/NVlabs/edm

Table 1: FID scores (lower is better) of different methods on low-shot datasets (256×256). The FIDs are means over three runs with all standard deviations less than 1% relatively.

Method	**Pre-training?**	**100-shot**			**Animal-Face**	
		Obama	Grumpy	Panda	Cat	Dog
EDM (Karras et al., 2022)	No	51.30	36.90	23.70	48.60	100.10
EDM + DA (Karras et al., 2022)	No	37.10	29.94	10.81	36.88	57.14
Patch Diffusion (Wang et al., 2023)	No	41.47	30.89	13.25	43.71	72.17
WILD-Diffusion (Wang et al., 2026)	No	34.52	26.33	8.96	34.21	53.18
LPDM-8 (Wang et al., 2023)	Yes	14.27	14.56	5.13	14.92	15.95
LD-Diffusion (Zhang et al., 2025)	Yes	13.00	13.31	4.70	12.77	12.48
WILD-Diffusion (Latent)(Wang et al., 2026)	Yes	12.54	12.83	4.66	12.93	12.21
Hybrid Diffusion	Yes	**10.92**	**9.99**	**3.71**	**11.13**	**11.26**

Table 2: A comparison of Precision (P) and Recall (R) of Hybrid Diffusion with other diffusion models on the 256×256 low-shot datasets. Massive Augmentation (MA) is applied to all of the methods. The Precision and Recall are means over three runs with all standard deviations less than 1% relatively.

Method	**Obama**		**Grumpy Cat**		**Panda**		**AF-Cat**		**AF-Dog**	
	P	R	P	R	P	R	P	R	P	R
EDM + DA	0.965	0.380	0.870	0.330	0.874	0.360	0.969	0.306	0.955	0.188
Patch Diffusion	0.952	0.410	0.861	0.370	0.866	0.392	0.961	0.324	0.943	0.201
LPDM-8	0.991	0.180	0.990	0.100	0.998	0.120	0.997	0.146	0.997	0.772
LD-Diffusion	0.996	0.100	0.994	0.080	**1.000**	0.012	0.999	0.103	1.000	0.773
Hybrid Diffusion	**0.998**	**0.840**	**1.000**	**0.830**	0.998	**0.890**	**1.000**	**0.894**	**1.000**	**0.902**

512×512, as used in existing studies (Karras et al., 2020a; Zhao et al., 2020; Li et al., 2022). We evaluate Hybrid Diffusion using FID (Heusel et al., 2017), CLIP-FID ($\mathrm{FID_{CLIP}}$) (Kynkäänniemi et al., 2023), and Precision/Recall (Kynkäänniemi et al., 2019; Sajjadi et al., 2018). The full dataset serves as the reference distribution, following prior work (Zhao et al., 2020; Karras et al., 2020a). The overall training duration is set to $20000\,\mathrm{kimg}$ for FFHQ and $40000\,\mathrm{kimg}$ for low-shot and BRECAHAD datasets for the latent stage in Hybrid Diffusion. A single Adam optimizer then updates both stages for the remainder of the training budget. To reduce the total computational cost, we evaluate the Hybrid Diffusion every $500\,\mathrm{kimg}$ for FFHQ and every $1000\,\mathrm{kimg}$ for low-shot and BRECAHAD datasets.

4.1 Results on Low-shot Datasets

We compare Hybrid Diffusion with both pixel and latent diffusion models on the low-shot datasets. The results are shown in Table 1. Hybrid Diffusion achieves the lowest FID compared with other diffusion models. Moreover, as shown in Tables 2 and 3, Hybrid Diffusion also achieves the comparable/higher Precision/Recall and lower $\mathrm{FID_{CLIP}}$ values compared with both pixel and latent diffusion models, which demonstrates the superiority of the proposed Hybrid Diffusion. To further demonstrate the effectiveness of Hybrid Diffusion, more generated images on low shot datasets are shown in Figure 3.

Table 3: Compared $\mathrm{FID_{CLIP}}$ scores between Hybrid Diffusion, LD-Diffusion and EDM + DA on low-shot datasets. We use the FID calculation model to calculate the $\mathrm{FID_{CLIP}}$. Here, O denotes Obama, GC denotes Grumpy Cat, P denotes Panda, and AF means AnimalFace.

Method	**100-shot**			**AF**	
	O	GC	P	Cat	Dog
EDM + DA (Karras et al., 2022)	3.81	5.45	3.93	8.29	10.06
LD-Diffusion (Zhang et al., 2025)	2.26	3.15	2.41	3.48	3.36
Hybrid Diffusion	**0.74**	**1.25**	**0.91**	**2.25**	**1.55**

Table 4: FID scores (lower is better) of different methods on the 256×256 FFHQ dataset. Massive Augmentation (MA) (Cui et al., 2022), i.e., xflipping, is applied in all methods. A dash indicates that the EDM run was not available in the supplied matched record. The FIDs are means over three runs with all standard deviations less than 1% relatively.

Method	**Pre-training?**	**FFHQ**		
		100	1K	2K
EDM (Karras et al., 2022)	No	79.10	-	-
EDM + DA (Karras et al., 2022)	No	50.73	30.75	27.17
Patch Diffusion (Wang et al., 2023)	No	44.45	28.03	25.32
LPDM-8 (Wang et al., 2023)	Yes	32.78	19.67	15.47
LD-Diffusion (Zhang et al., 2025)	Yes	**28.51**	17.76	14.36
Hybrid Diffusion	Yes	31.24	**16.95**	**14.11**

Table 5: Precision (P, higher is better), Recall (R, higher is better), and CLIP-FID (C-FID, lower is better) on the 256×256 FFHQ dataset. The results are means over three runs with all standard deviations less than 1% relatively.

Method	FFHQ-100			FFHQ-1K			FFHQ-2K		
	P	R	C-FID	P	R	C-FID	P	R	C-FID
EDM + DA	0.724	0.005	41.15	0.668	0.222	23.92	0.709	0.281	15.31
LD-Diffusion	**0.791**	0.018	12.75	0.767	0.249	6.87	0.769	0.323	5.32
Hybrid Diffusion	0.787	**0.044**	**11.45**	**0.793**	**0.311**	**3.67**	**0.809**	**0.412**	**3.15**

4.2 Results on FFHQ Datasets

We compare Hybrid Diffusion with both pixel and latent diffusion models on the 256×256 FFHQ dataset under limited-data settings. The results are reported in Tables 4 and 5. Hybrid Diffusion achieves comparable FID and Precision on 100 subsets and the lowest FID and highest Precision and Recall on both the 1K and 2K subsets. Moreover, as shown in Table 5, it obtains the best FID_{CLIP} values among all compared methods, demonstrating the superiority of the proposed Hybrid Diffusion in learning high-quality image distributions from limited training data. More generated images of Hybrid Diffusion on FFHQ datasets can be found in Figure 4 (a)-(c).

4.3 Results on the BRECAHAD Dataset

We further evaluate Hybrid Diffusion on the 512×512 BRECAHAD dataset, a domain-specific medical imaging dataset. The results in Table 6 demonstrate that Hybrid Diffusion consistently outperforms existing pixel-space and latent-space diffusion methods, confirming its effectiveness in specialized medical imaging scenarios. In particular, as shown in Figure 4 (d), the high generation quality indicates that Hybrid Diffusion better preserves fine-grained visual details, such as tissue structures and cell boundaries, which are critical for high-fidelity medical image synthesis.

4.4 Ablation Studies

We conduct a series of ablation studies on the 100-shot Obama dataset to investigate the contributions of the key design choices in Hybrid Diffusion. The detailed results are presented below.

Conditioning source and gradient flow. To show the effectiveness of the conditioning strategy in Hybrid Diffusion, we conduct an ablation study by varying both the source of the pixel-space condition and whether the pixel stage loss is propagated into the latent denoiser. The results are shown in Table 7. Conditioning on the clean latent reconstruction $g(z_0)$ introduces a train–test discrepancy, since the pixel denoiser never sees the imperfect decoded predictions produced at inference time. Allowing the pixel loss to back-propagate through the latent stage couples the two optimization objectives and degrades performance, whereas model-produced conditioning exposes the pixel denoiser to latent-prediction errors during training. In contrast, the proposed Hybrid Diffusion uses model-produced conditioning with a stop-gradient, so the latent denoiser retains its original

Table 6: FID scores of different diffusion models on the BRECAHAD dataset. Precision, Recall, and CLIP-FID were not evaluated in the supplied record. The FIDs are means over three runs with all standard deviations less than 1% relatively.

Method	Latent?	FID
EDM + DA (Karras et al., 2022)	No	116.04
LD-Diffusion (Zhang et al., 2025)	Yes	30.62
Hybrid Diffusion	Yes	**21.57**

Table 7: Ablation of the conditioning strategy on the 100-shot Obama dataset. We vary both the source of the pixel space condition and whether the pixel stage loss is propagated into the latent denoiser.

Conditioning Strategy	**Gradient to Latent**	**FID↓**
Clean-latent reconstruction $g(z_0)$	No	11.79
Clean-latent reconstruction $g(z_0)$	Yes	13.76
Model-produced condition	Yes	12.36
Model-produced condition	No	**10.92**

objective. Among the evaluated configurations, this setting achieves the best FID, supporting the importance of both model-produced conditioning and decoupled optimization.

Latent-to-pixel transition point. The latent-to-pixel transition is designed to balance global structure formation in latent space with fine-detail refinement in pixel space. The switch level σ_{switch} controls how the denoising trajectory is allocated between these two stages: an earlier transition assigns more of the trajectory to pixel-space denoising, while a later transition allows more denoising to be performed in latent space. To examine the effect of this allocation, we evaluate different switch levels on the 100-shot Obama dataset. As shown in Table 8, σ_{switch}=0.05 achieves the lowest FID of 10.92.

Table 8: Ablation of the latent-to-pixel transition level σ_{switch} on the 100-shot Obama dataset. The switch level controls the allocation of denoising between latent and pixel spaces.

σ_{switch}	0.025	0.035	0.05	0.075	0.10	0.15
FID↓	11.07	11.02	**10.92**	10.99	11.04	11.06

Pixel-space initialization. We evaluate pixel-space initialization on the 100-shot Obama dataset by comparing deterministic initialization, $x_{t_{\text{cur}}} = c_s$ ($\eta = 0$), with stochastic initialization, $x_{t_{\text{cur}}} = c_s + \eta\epsilon_x$ ($\eta = t_{\text{cur}}$), while keeping all other components fixed. As shown in Table 9, stochastic initialization improves the FID from 14.01 to 10.92, indicating that switch-scale noise provides a local neighborhood for residual correction while preserving the global structure.

Table 9: Pixel stage initialization of Hybrid Diffusion on the 100-shot Obama dataset.

Initialization	$\boldsymbol{\eta}$	**FID↓**
Deterministic: $x_{t_{\text{cur}}} = c_s$	0	14.01
Stochastic: $x_{t_{\text{cur}}} = c_s + \eta\epsilon_x$	t_{cur}	**10.92**

5 Conclusion

We introduced Hybrid Diffusion, a hybrid latent-to-pixel diffusion framework for limited-data generation that retains the efficiency of compact latent modeling while alleviating the fidelity constraints imposed by a frozen autoencoder. Hybrid Diffusion performs most of the generative trajectory in latent space and transfers the process to a conditional pixel denoiser at the hard sampling switch. Experiments on several commonly-used datasets demonstrate the effectiveness of Hybrid Diffusion.

AI USE STATEMENT

We used generative AI tools extensively to assist with writing and language polishing, as well as with proving mathematical claims. For writing, these tools helped revise and polish the manuscript to improve clarity and readability. For mathematical proofs, they were used to propose proof strategies and to generate and revise candidate proofs.

The authors independently checked the final proofs step by step, examining the logical validity of the arguments and their consistency with the stated assumptions. The authors take full responsibility for the final content of this work, including the text, mathematical claims, and proofs developed with the assistance of generative AI.

REFERENCES

Alper Aksac, Douglas J Demetrick, Tansel Ozyer, and Reda Alhajj. Brecahad: a dataset for breast cancer histopathological annotation and diagnosis. *BMC research notes*, 12:1–3, 2019.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov):463–482, 2002.

Nicholas M Boffi, Arthur Jacot, Stephen Tu, and Ingvar Ziemann. Shallow diffusion networks provably learn hidden low-dimensional structure. In *ICLR*, 2024.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security*, 2023.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *ICML*, 2023.

Kaiwen Cui, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, Fangneng Zhan, and Shijian Lu. Genco: Generative co-training for generative adversarial networks with limited data. In *AAAI*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

Yunxiang Fu, Chaoqi Chen, Yu Qiao, and Yizhou Yu. Dreamda: Generative data augmentation with diffusion models. *arXiv preprint arXiv:2403.12803*, 2024.

Giorgio Giannone, Didrik Nielsen, and Ole Winther. Few-shot diffusion models. *arXiv preprint arXiv:2205.15463*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of styleGAN. In *CVPR*, 2020b.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *ICLR*, 2023.

Ziqiang Li, Chaoyue Wang, Heliang Zheng, Jing Zhang, and Bin Li. Fakeclr: Exploring contrastive learning for solving latent discontinuity in data-efficient gans. In *ECCV*, 2022.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40, 2024. doi: 10.1007/978-3-031-72980-5_2.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Taehong Moon, Moonseok Choi, Gayoung Lee, Jung-Woo Ha, and Juho Lee. Fine-tuning diffusion models with limited data. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-decoding models for few-shot conditional generation. In *NeurIPS*, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *NeurIPS*, 2021.

Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

Xianglu Wang, Wanlin Zhang, and Hu Ding. Wild-diffusion: A wdro inspired training method for diffusion models under limited data. In *ICLR*, 2026.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. In *NeurIPS*, 2023.

Ruofeng Yang, Bo Jiang, Cheng Chen, Ruinan Jin, Baoxiang Wang, and Shuai Li. Few-shot diffusion models escape the curse of dimensionality. In *NeurIPS*, 2024.

Zhaoyu Zhang, Yang Hua, Guanxiong Sun, Hui Wang, and Seán McLoone. Training diffusion-based generative models with limited data. In *ICML*, 2025.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. In *NeurIPS*, 2020.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026.

Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. Few-shot image generation with diffusion models. *arXiv preprint arXiv:2211.03264*, 2022.

A Additional Analysis of Hybrid Diffusion

A.1 Finite-sample denoising analysis via ERM

At noise level σ_t, let $\boldsymbol{n}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I})$. For a clean sample $\boldsymbol{y}$, define the denoising loss as

$$\ell_t(D; \boldsymbol{y}) \coloneqq \mathbb{E}_{\boldsymbol{n}_t}\left[\|D(\boldsymbol{y} + \boldsymbol{n}_t; \sigma_t) - \boldsymbol{y}\|_2^2\right]. \tag{17}$$

Let $\boldsymbol{y}_i \overset{\text{i.i.d.}}{\sim} P_{\text{data}}$ and $S_N = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ denote the training set. The population and empirical risks are

$$\mathcal{R}_t(D) \coloneqq \mathbb{E}_{\boldsymbol{Y} \sim P_{\text{data}}}\left[\ell_t(D; \boldsymbol{Y})\right], \qquad \widehat{\mathcal{R}}_{t,S_N}(D) \coloneqq \frac{1}{N}\sum_{i=1}^{N} \ell_t(D; \boldsymbol{y}_i). \tag{18}$$

An empirical risk minimizer over a hypothesis class $\mathcal{D}$ is

$$\widehat{D}_{S_N} \in \arg\min_{D \in \mathcal{D}} \widehat{\mathcal{R}}_{t,S_N}(D).$$

Let $\mathcal{D}_{\text{all}}$ denote the class of measurable, square-integrable denoisers, and define

$$\mathcal{R}_t^\star \coloneqq \inf_{D \in \mathcal{D}_{\text{all}}} \mathcal{R}_t(D).$$

This infimum is attained by the MMSE denoiser

$$D_t^{\text{MMSE}}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{Y} \mid \boldsymbol{Y} + \boldsymbol{n}_t = \boldsymbol{x}].$$

Let

$$D_{\mathcal{D}}^\star \in \arg\min_{D \in \mathcal{D}} \mathcal{R}_t(D).$$

Then the expected population risk of ERM admits the decomposition

$$\begin{aligned} \mathcal{E}_t(\mathcal{D}, N) &\coloneqq \mathbb{E}_{S_N \sim P_{\text{data}}^N}\left[\mathcal{R}_t(\widehat{D}_{S_N})\right] \\ &= \underbrace{\mathcal{R}_t^\star}_{\text{irreducible (MMSE) risk}} + \underbrace{\left[\mathcal{R}_t(D_{\mathcal{D}}^\star) - \mathcal{R}_t^\star\right]}_{\mathcal{E}_{\text{app}}(\mathcal{D})} + \underbrace{\mathbb{E}_{S_N}\left[\mathcal{R}_t(\widehat{D}_{S_N}) - \mathcal{R}_t(D_{\mathcal{D}}^\star)\right]}_{\mathcal{E}_{\text{est}}(\mathcal{D}, N)}. \end{aligned} \tag{19}$$

For fixed P_{data}, σ_t, and hypothesis class $\mathcal{D}$, only the estimation term depends on N. Under standard conditions such as bounded data and outputs, or suitable sub-Gaussian tail assumptions, ERM theory gives

$$\mathcal{E}_{\text{est}} \lesssim \mathfrak{R}_N(\ell_t \circ \mathcal{D}) \lesssim \sqrt{\frac{\mathcal{C}_t(\mathcal{D})}{N}}, \tag{20}$$

where $\mathfrak{R}_N(\ell_t \circ \mathcal{D})$ denotes the Rademacher complexity of the loss class and $\mathcal{C}_t(\mathcal{D})$ is a problem-dependent effective complexity measure.

This decomposition suggests why latent diffusion can be advantageous in limited-data regimes. Because the effective complexity depends on the denoiser parameterization and input/output dimensions, operating in a compact latent space may reduce $\mathcal{C}_t(\mathcal{D})$ and hence the estimation error. This argument assumes that the encoder and decoder are fixed, or that their complexity is included in $\mathcal{C}_t(\mathcal{D})$, and that the associated representation and reconstruction errors remain controlled. These errors contribute to the approximation term $\mathcal{E}_{\text{app}}$.

A.2 Decomposing representation and decoded prediction residuals

For the decoded model condition $c_c = g(\hat{z}_{0,c})$, adding and subtracting $g(z_0)$ gives

$$\begin{aligned} r_{\text{rep}} &= x_0 - g(z_0), \\ r_{\text{pred}} &= g(z_0) - g(\hat{z}_{0,c}), \\ x_0 - c_c &= r_{\text{rep}} + r_{\text{pred}}. \end{aligned} \tag{21}$$

Here, r_{rep} is the reconstruction error obtained by encoding and decoding the true latent z_0. In contrast, r_{pred} is the pixel space error induced by replacing z_0 with the predicted latent $\hat{z}_{0,c}$; it is therefore a decoded prediction residual rather than the latent prediction error itself. The triangle inequality gives Equation (4), and if g is L_g-Lipschitz under the stated norms, then

$$\|x_0 - c_c\|^2 \le 2\|r_{\mathrm{rep}}\|^2 + 2L_g^2\|z_0 - \hat{z}_{0,c}\|^2,$$

which yields Equation (5).

These identities characterize the conditioning error before refinement. If the condition is $g(z_0)$, its discrepancy from x_0 is exactly r_{rep}. If the condition is $c_c = g(\hat{z}_{0,c})$, its discrepancy is $r_{\mathrm{rep}} + r_{\mathrm{pred}}$, where r_{pred} denotes the additional image-space error caused by replacing the true latent z_0 with the predicted latent $\hat{z}_{0,c}$. Thus, when c_c is used as the input to the pixel-space branch, the branch receives a condition affected by both residual sources, whereas using $g(z_0)$ exposes it only to the representation residual. This is an exact per-sample error decomposition together with an upper bound; the two residuals may partially cancel, and the decomposition alone does not imply residual removal or improved downstream metrics.

The two decoded conditions also lie in $\mathrm{range}(g)$ by construction. In contrast, a refiner R_ϕ that outputs directly in pixel space is not constrained to this range. It can therefore, in principle, correct decoder-specific errors or synthesize details that the decoder cannot represent. This describes the expressive capacity of the refiner, rather than guaranteeing that such corrections or details will be recovered.

A.3 Alignment of training and sampling conditions

The conditioning signals supplied to the pixel stage are constructed from different latent states during training and sampling. During training, the condition in Equation (8) is obtained by applying the latent predictor to a single forward-corrupted version of the clean latent at noise level σ_c:

$$c_c = g(D_\theta(z_0 + \sigma_c\epsilon_c, \sigma_c)), \qquad \bar{c}_c = \mathrm{sg}(c_c).$$

During sampling, the latent stage is first evolved to the hard switch, and the condition in Equation (15) is computed from the state produced by this trajectory:

$$c_s = g(D_\theta(z_s, t_{\mathrm{cur}})).$$

Thus, c_c and c_s are not expected to be numerically identical. Their alignment is functional: both are generated by the same decoded-prediction map $g \circ D_\theta$ and serve as the condition for the pixel stage. The forward-corrupted construction provides a tractable stochastic surrogate for the condition encountered near the switch, exposing the pixel stage to errors in latent prediction without requiring an unrolled latent trajectory for every training update. In the pixel loss, we use the detached condition $\bar{c}_c$. This leaves its numerical value unchanged while blocking gradients from the pixel objective from propagating through the condition-construction path into the latent stage.

A.4 Stochastic initialization and conditional refinement

During training, let $p_0(dx_0 \mid c_c)$ denote the conditional distribution of clean targets. Perturbing x_0 with Gaussian noise of scale σ_p gives the conditional training marginal

$$p_{\sigma_p}(x \mid c_c) = \int \mathcal{N}\big(x; x_0, \sigma_p^2 I\big)\; p_0(dx_0 \mid c_c). \tag{22}$$

At inference, Eq. (16) initializes the state as

$$x_{t_{\mathrm{cur}}} = c_s + t_{\mathrm{cur}}\epsilon_x, \qquad \epsilon_x \sim \mathcal{N}(0, I). \tag{23}$$

Therefore, conditioned on c_s, the state entering the refinement stage has distribution

$$q_{t_{\mathrm{cur}}}(x \mid c_s) = \mathcal{N}\big(x; c_s, t_{\mathrm{cur}}^2 I\big). \tag{24}$$

These two distributions generally do not coincide. The training marginal is a Gaussian convolution, or equivalently a mixture of Gaussians, over clean targets conditioned on c_c, whereas the inference-time initialization is a single Gaussian centered at c_s. The conditioning signals and the variables

being corrupted are also constructed differently. Consequently, matching t_{cur} with σ_p alone does not guarantee that the two distributions are equal.

We therefore interpret the switch as an approximate conditional-refinement interface between the inference-time initialization distribution and the conditional denoising distribution learned during training. The adequacy of this approximation is evaluated empirically.

A.5 Residual Learning and Statistical Learnability

The residual identity above specifies the learning target of the pixel stage, while the finite-sample bound below characterizes its statistical cost. Under standard learning-theoretic assumptions, for a fixed conditional denoising problem with hypothesis class $\mathcal{H}_{\mathrm{cond}}$,

$$\mathcal{E}_{\mathrm{est}}^{\mathrm{cond}} \lesssim \sqrt{\frac{\mathcal{C}(\mathcal{H}_{\mathrm{cond}})}{N}}. \tag{25}$$

Here, $\mathcal{C}(\mathcal{H}_{\mathrm{cond}})$ denotes an effective complexity measure, up to constants and logarithmic factors. Conditioning, a restricted pixel-noise interval, and a compact refiner are intended to control this complexity, but they do not by themselves establish a smaller value of $\mathcal{C}(\mathcal{H}_{\mathrm{cond}})$. The hybrid design can therefore be beneficial insofar as the conditional residual problem is learnable from the available data; whether this holds is ultimately an empirical question.

B Gradient Isolation under Coupled Training

The stop-gradient in Equation (10) gives the following result for the joint objective in Equation (13). Let a_k indicate whether pixel training is active at update k:

$$\mathcal{L}_k = \mathcal{L}_{\mathrm{lat}}(\theta) + a_k \lambda_p \mathcal{L}_{\mathrm{pix}}(\phi; \mathrm{sg}(c_\theta)), \tag{26}$$

where $c_\theta = g(D_\theta(z_c, \sigma_c))$ and θ, ϕ are disjoint.

Proposition 1 (Isolation of the latent gradient)**.** *If the pixel loss depends on θ only through $\mathrm{sg}(c_\theta)$, and no other pixel input, target, weight, or gate depends differentiably on θ, then*

$$\nabla_\theta \mathcal{L}_{\mathrm{pix}} = 0, \qquad \nabla_\theta \mathcal{L}_k = \nabla_\theta \mathcal{L}_{\mathrm{lat}}, \qquad \nabla_\phi \mathcal{L}_k = a_k \lambda_p \nabla_\phi \mathcal{L}_{\mathrm{pix}}. \tag{27}$$

Proof. Stop-gradient has identity forward value and zero backward Jacobian. For a sampled pixel loss,

$$\begin{aligned} \nabla_\theta \ell_{\mathrm{pix}} &= J_{c_\theta,\theta}^{\mathsf{T}} J_{\mathrm{sg},c_\theta}^{\mathsf{T}} \nabla_{\mathrm{sg}(c_\theta)} \ell_{\mathrm{pix}} \\ &= J_{c_\theta,\theta}^{\mathsf{T}} 0 = 0. \end{aligned} \tag{28}$$

The same identity holds for a minibatch average, which proves the stated gradients. □

The proposition concerns the backward computation at a fixed parameter state. The forward condition still changes with θ, so the two branches remain coupled through their inputs. Exact equality of latent training trajectories would require matching the optimizer state, random draws, schedules, and all other stateful operations.

C Extension to a Velocity-Based Latent Model

The interface used in Table 1 requires a clean-latent estimate at the switch, regardless of whether the latent model predicts clean data or velocity. For the affine path

$$z_t = a(t) z_0 + b(t)\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{29}$$

the optimal squared-error velocity predictor is $v^*(z_t, t) = \mathbb{E}[\dot{a}(t) z_0 + \dot{b}(t)\epsilon \mid z_t]$. Whenever $a\dot{b} - \dot{a}b \neq 0$,

$$\mathbb{E}[z_0 \mid z_t] = \frac{\dot{b}(t) z_t - b(t) v^*(z_t, t)}{a(t)\dot{b}(t) - \dot{a}(t) b(t)}. \tag{30}$$

Replacing v^* with v_ψ gives the clean estimate used by the interface. Under the data-to-noise convention $a(t) = 1 - t$, $b(t) = t$,

$$\hat{z}_0(z_t, t) = z_t - t\, v_\psi(z_t, t). \tag{31}$$

For the opposite time convention, the sign and time coordinate must be transformed consistently. At a sampled training time t_c, the detached condition is

$$\bar{c}_c = \mathrm{sg}(g(z_c - t_c v_\psi(z_c, t_c))), \qquad z_c = (1 - t_c)z_0 + t_c \epsilon_c. \tag{32}$$

At inference, the same conversion is applied at the SiT switch before decoding and pixel refinement. SiT time is separate from the EDM pixel-noise coordinate.

D Training and Sampling Procedures

The following steps expand the objective in Equation (13) and the hard handoff in Section 3.2.4.

D.1 Generation with a fixed decoded condition

Sampling follows the shared EDM grid to σ_{switch}. Decode the latent clean prediction once to obtain c_s, initialize with Equation (16), and keep c_s fixed during the remaining pixel trajectory. For positive levels $\sigma_i > \sigma_{i+1} > 0$, define

$$d_\phi(x, c, \sigma) = \frac{x - R_\phi(x, c, \sigma)}{\sigma}. \tag{33}$$

A Heun update is

$$\begin{aligned} h_i &= \sigma_{i+1} - \sigma_i, \qquad d_i = d_\phi(x_i, c_s, \sigma_i), \\ \widetilde{x}_{i+1} &= x_i + h_i d_i, \quad x_{i+1} = x_i + \frac{h_i}{2}\left[d_i + d_\phi(\widetilde{x}_{i+1}, c_s, \sigma_{i+1})\right]. \end{aligned} \tag{34}$$

At the final transition to zero, use $x_{\mathrm{out}} = R_\phi(x_i, c_s, \sigma_i)$. The exact grid, churn settings, and trained pixel-noise interval are configuration choices and should be recorded with the run.

D.2 Counting denoiser evaluations

Let n_{lat} count latent-network evaluations, excluding a separate call used only to construct the switch condition, and let n_{pix} count pixel-network evaluations. Then

$$\mathrm{NFE}_{\mathrm{hybrid}} = n_{\mathrm{lat}} + \delta_{\mathrm{extra}} + n_{\mathrm{pix}}, \qquad \delta_{\mathrm{extra}} \in \{0, 1\}. \tag{35}$$

Decoder calls are excluded from denoiser NFE but contribute to end-to-end latency. Because the branches can have different resolutions and architectures, branch NFE must be reported together with a common-device timing protocol.

E More experimental results

E.1 Generalization Ability to Other Diffusion Backbones

To further assess the backbone generalization of Hybrid Diffusion, we implement it with the SiT backbone (Ma et al., 2024). We refer to this variant as *Hybrid Diffusion (SiT)*. Table 11 reports the FID scores at 256×256 resolution on five low-shot datasets, including three 100-shot datasets (Obama, Grumpy, and Panda) and two Animal-Face datasets (Cat and Dog). All methods use pre-training, and each FID score is averaged over three independent runs, with a relative standard deviation below 1% in all settings.

As shown in Table 11, Hybrid Diffusion (SiT) consistently outperforms the SiT-XS (latent) baseline across all five datasets. In particular, it achieves lower FID scores on both the 100-shot and Animal-Face datasets, demonstrating that the benefits of Hybrid Diffusion are not tied to a specific diffusion backbone. These results suggest that Hybrid Diffusion can be effectively transferred to other diffusion architectures.

Table 10: Common protocol for all ablation studies. Unless explicitly varied in a given ablation, the settings below are fixed across the compared variants.

Protocol component	Fixed setting across ablations
Scope and data	100-shot Obama at 256×256; the training subset, preprocessing, augmentation, and random-seed protocol are matched across all variants.
Frozen representation	Frozen/evaluation-mode `stabilityai/sd-vae-ft-mse`, posterior-mode encoding, latent scale $s = 0.18215$, and $4 \times 32 \times 32$ latents (equivalently $4 \times (H/8) \times (W/8)$).
Networks	The latent and pixel networks are fixed throughout, with 63,715,847 parameters in total. Pixel-space forward passes use FP32.
Optimization and budget	One Adam optimizer; the total training budget and evaluation interval are fixed. The default pixel-loss weight is $\lambda_p = 1$.
Training noise	Latent $\sigma = \exp(\mathcal{N}(-1.2, 1.2^2))$ with $\sigma_{\text{data}} = 0.5$; pixel σ_p is sampled log-uniformly from $[0.002, 0.05]$.
Condition and handoff	The default condition is model-produced and detached before the pixel loss. A hard latent–pixel handoff is made on one shared EDM noise grid at σ_{switch}, with $x_{t_{\text{cur}}} = c_s + t_{\text{cur}}\epsilon_x$. The default start is stochastic; the initialization ablation compares it with $\eta = 0$.
Sampling and evaluation	Heun EDM sampling with the same solver grid and sampling seeds is used for every variant. Checkpoints are selected by FID from the complete latent-to-pixel sampler against the declared full-dataset reference distribution. FID is the primary ablation metric; Precision/Recall and CLIP-FID, when reported, use the same reference and are listed separately.
Controlled factors	Each comparison changes only its designated factor(s) while the remaining settings are fixed: condition source and/or gradient flow, σ_{switch}, pixel-stage initialization.

Table 11: FID scores (lower is better) on low-shot datasets at 256×256 resolution. Each score is averaged over three independent runs, with a relative standard deviation below 1% for all settings.

Method	**Pre-trained?**	**100-shot**			**Animal-Face**	
		Obama	Grumpy	Panda	Cat	Dog
SiT-XS (latent) (Ma et al., 2024)	Yes	12.06	13.47	4.46	11.87	19.26
Hybrid Diffusion (SiT)	Yes	**9.94**	**9.67**	**3.40**	**9.83**	**18.21**

E.2 More generated results

More generated images produced by Hybrid Diffusion are shown in Figures 3 and 4.

E.3 Computational cost

The comparisons of the inference time of EDM + DA, LD-Diffusion and Hybrid Diffusion on the 100-shot Obama dataset have been demonstrated in Table 12. Although LD-Diffusion is approximately 12.5% faster than Hybrid Diffusion, the proposed Hybrid Diffusion still reduces the infer-

Method	**Generating 5K images during inference (min)**
EDM + DA	165
LD-Diffusion	28
Hybrid Diffusion	32

Table 12: The inference time of EDM + DA, LD-Diffusion and Hybrid Diffusion on the 100-shot Obama dataset (256×256). The half precision, i.e., FP16, is applied to all methods. The results are calculated by averaging over ten times on two NVIDIA A5000 GPUs. All standard deviations are less than 1%, relatively.

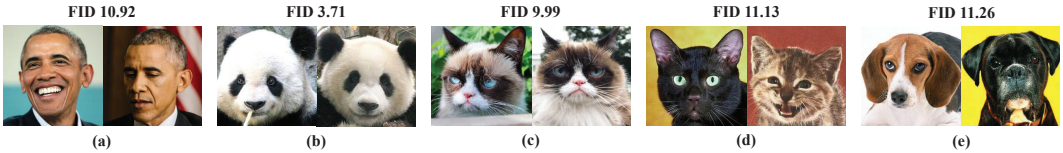


Figure 3: The generated images by Hybrid Diffusion on (a) 100-shot Obama, (b) 100-shot Panda, (c) 100-shot Grumpy-cat, (d) AnimalFace-cat and (e) AnimalFace-dog datasets.

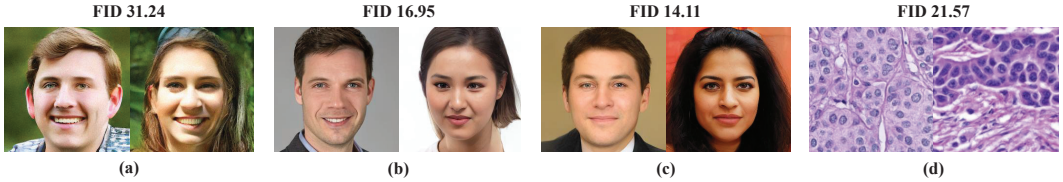


Figure 4: The generated images by Hybrid Diffusion on (a) FFHQ-100 (b) FFHQ-1000 (c) FFHQ-2000 (d) Brecahad.

ence time by approximately 80.6% compared with EDM + DA. This indicates that Hybrid Diffusion achieves a computational cost close to that of the most efficient baseline while retaining the quality advantages reported above, demonstrating a favorable efficiency–quality trade-off.

F Interpretation of the Controlled Ablations

The ablations in Section 4.4 use the 100-shot Obama setting while holding the backbone, budget, and evaluation protocol fixed as shown in Table 10.

G Broader Impact and Reproducibility

Broader impact. This paper focuses on diffusion models with limited data. The technical contributions of this paper do not raise any particular ethical challenges. However, because technology is usually a double-edged sword, our work may also bring potential social risks when applying diffusion models with limited data. For example, it may ease the high-quality fake media synthesis using only limited data.

Reproducibility. Exact reproduction of each reported number requires the settings in Table 10, the checkpoint-selection rule, code and model versions, dataset manifests, reference statistics, and training and sampling seeds. The run manifest should additionally record preprocessing and augmentation, the frozen autoencoder revision and latent scaling, network architectures, optimizer and learning-rate schedule, batch size and precision and total budget, latent and pixel noise laws, the σ_{switch} value, the complete sampler grid and solver settings, the number of generated samples, metric implementations and feature-model versions, and the hardware/software environment. The mathematical description alone does not determine the optimizer state, data order, random-number streams, numerical kernels, or checkpoint selected by the pipeline, so it does not imply bitwise-deterministic reproduction across environments.