Masked Generative Pretraining Improves Cross-Dataset Transfer in Pixel-Space Diffusion

Shu Wei* **Jiachen Lei** **Jiahong Wu†** **Xiangxiang Chu**
AMAP, Alibaba Group

Abstract

Pixel-space diffusion models have recently seen a revival in image generation, yet their training remains less efficient than that of their latent-space counterparts. Existing methods reduce this cost by equipping diffusion models with external representations from clean images. In contrast, representation consistency training, a new generative pretraining paradigm, stands out as it eliminates the need for any off-the-shelf semantic encoders. It proposes pretraining the diffusion model to simultaneously capture meaningful visual semantics from clean images while aligning them with data points across various noise levels. Nevertheless, its training depends on contrastive learning with hand-crafted augmentations. These augmentations impose predefined invariances during pretraining, potentially limiting model performance when transferred to downstream target distributions that substantially deviate from the pretraining manifold. To address this, we propose Masked Pixel-space Generative Pretraining (MPG), an augmentation-free framework based on masked image modeling. MPG trains the encoder to recover masked image regions and aligns the resulting predictions across different noise levels. Under the same architectures and fine-tuning settings, MPG produces better generation performance across seven downstream datasets, while retaining competitive generation quality on ImageNet-256. Further analyses show that MPG exhibits weaker invariance to hand-crafted augmentations and better preserves its pretrained representation geometry during downstream fine-tuning. These results suggest masked generative pretraining as a practical alternative for training pixel diffusion models.

1 Introduction

Driven by substantial improvements in training efficiency, pixel-space diffusion models [Li and He, 2025, Chen et al., 2025, Hoogeboom et al., 2025, 2023, Lei et al., 2025] have experienced a significant revival. While some recent works focus on exploring novel architectures objectives [Ma et al., 2025, Yu et al., 2025b, Wang et al., 2025] or intermediate representation supervision [Yu et al., 2025b, Ma et al., 2025, Wang et al., 2025, Lei et al., 2025], others [Lei et al., 2025] demonstrate the efficacy of pretraining diffusion models by enforcing representation consistency across noise levels before fine-tuning diffusion models under standard denoising training settings. The pretraining is termed as Representation Consistency Training. This pretrain-finetune paradigm allows a single pretrained encoder to be repeatedly fine-tuned for various downstream generative tasks, significantly improving both convergence speed and generation performance.

However, its contrastive pretraining objective relies heavily on data pairs constructed through hand-crafted augmentations, which impose predefined invariances on the learned representations. Unlike discriminative representation learning, generative modeling requires not only high-level semantics

*Work done during an internship at AMAP.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

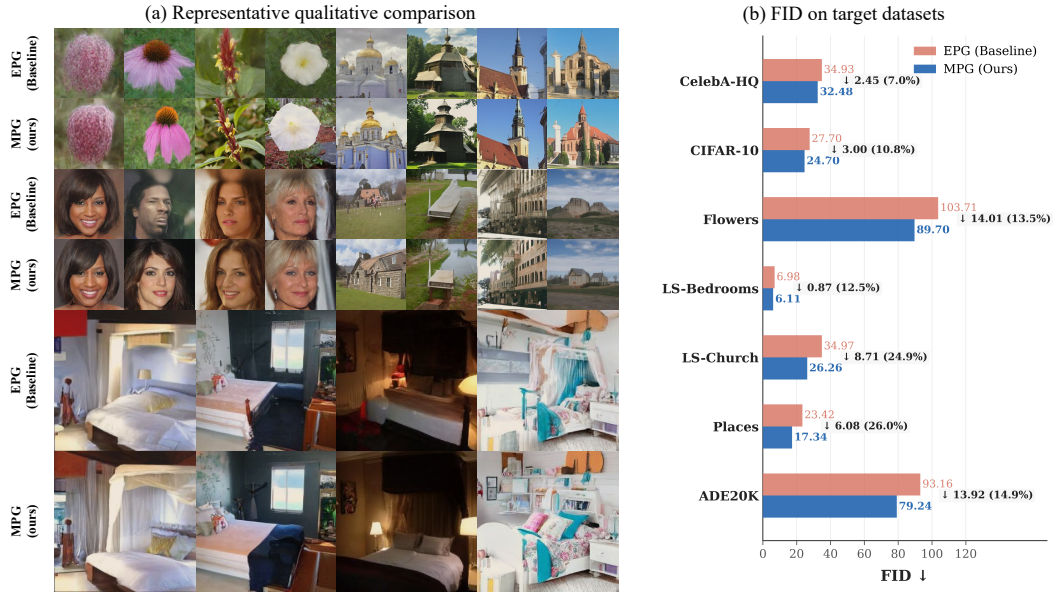

Figure 1: **MPG enables robust cross-dataset transfer for pixel-space diffusion.** (a) Representative qualitative comparisons between EPG and MPG across diverse target domains. Note: Images are scaled down for layout purposes. (b) FID comparison on all seven target datasets.

but also fine-grained visual appearance, such as texture and color. These augmentation-induced invariances may suppress visual details important for downstream generation, particularly when target distributions differ substantially from the pretraining distribution [He et al., 2020, Chen et al., 2020, Lee et al., 2021].

To address this limitation, we revisit the pretraining objective itself and propose Masked Pixel-space Generative Pretraining (MPG). During pretraining, the model learns to reconstruct masked image regions while enforcing consistent clean-image predictions across adjacent noise levels using masked noisy samples that share the same underlying image and noise realization. This dual objective encourages the model to capture low-level image details, while ensuring representation consistency along the PF ODE trajectory. During generative fine-tuning, we discard the lightweight decoder for masked image modeling and replace it with a randomly initialized higher-capacity decoder, while retaining the pretrained encoder. To validate this design, we pretrain our model on ImageNet-256 [Deng et al., 2009] and conduct extensive transferring experiments on seven downstream datasets that exhibit significant distribution shifts from the pretraining domain. As summarized in Figure 1, these results demonstrate the superior transferability of MPG.

Meanwhile, to better understand the representations underlying MPG's improved transferability, we conduct complementary analyses of augmentation invariance and representation changes during downstream fine-tuning. Using patch-level CKNNA, we find that MPG exhibits weaker invariance to hand-crafted augmentations and better preserves its pretrained representation geometry across seven downstream datasets compared with augmentation-based pretraining. To further characterize the pretraining dynamics, we introduce Noise-Consistency CKNNA (NC-CKNNA) as a diagnostic complementary to linear probing, evaluating the consistency of local representation geometry across noise levels. Together, these analyses provide further insights into the representation properties associated with MPG's improved transferability.

Our contributions are summarized as follows:

- We propose MPG, an augmentation-free framework that combines masked image modeling with representation consistency training that significantly benefits model transferability.
- We analyze MPG's transferability through augmentation invariance and pretrain-to-finetune representation drift, and introduce NC-CKNNA as an auxiliary diagnostic of representation consistency across noise levels.

- Through rigorous controlled evaluations across seven diverse downstream datasets, we demonstrate that MPG significantly improves transfer performance by eliminating the need of hand-crafted augmentation design. Furthermore, alongside these gains, MPG delivers competitive source-domain generation performance on ImageNet-256.

2 Related Works

Pixel-space diffusion models. Recent work has revived pixel-space generation and narrowed the efficiency and performance gap against latent-space diffusion models [Rombach et al., 2022] by improving training recipes [Chen, 2023, Kingma and Gao, 2023, Hoogeboom et al., 2023, 2025, Yu et al., 2026] and developing efficiency-oriented architectures. Innovations range from computation-reducing designs like RINs [Jabri et al., 2023] and the dual-path PixelDiT [Yu et al., 2025b], to detail-preserving mechanisms such as the neural-field heads in PixNerd [Wang et al., 2025] and frequency decoupling in DeCo [Ma et al., 2025]. The landscape is further diversified by hybrid flow-based or autoregressive formulations [Tschannen et al., 2024, Chen et al., 2025, Zheng et al., 2025, Xu et al., 2025, Jin et al., 2025]. Li and He [2025] show that even a minimalist large-patch image Transformer predicting clean pixels can be surprisingly strong without auxiliary losses or pretraining. This design philosophy we also adopt in this work. These advances demonstrate that pixel-space generation is a viable alternative to latent diffusion. However, most of them are trained from scratch for a target distribution and primarily optimize source-domain sample quality. In contrast, our work studies a complementary question: whether a self-supervised Stage-I objective can produce an encoder that transfers more reliably when the downstream generative dataset changes.

Representation learning for diffusion generation. A growing line of work shows that diffusion training can benefit from explicit representation learning [Peebles and Xie, 2023, Ma et al., 2024]. Yu et al. [2025a] and Leng et al. [2025] align diffusion features with representations from external self-supervised encoders [Caron et al., 2021], thereby improving convergence and generation quality. REPA-E [Leng et al., 2025] further extends this philosophy to an end-to-end framework by jointly optimizing the VAE and diffusion model via representation alignment for improved efficiency, while VA-VAE [Yao et al., 2025] aligns the tokenizer latent space with pretrained vision features to better structure high-dimensional latents, improving both reconstruction and downstream diffusion convergence. Beyond generative training, pretrained diffusion models have also been leveraged as priors for downstream image restoration tasks, as demonstrated by LD-RPS [Li et al., 2025]. In contrast to methods relying on external teachers, Stoica et al. [2025] and Wang and He [2025] impose auxiliary contrastive losses on predicted clean images or intermediate features, enforcing semantic structure without separate pretrained models. Lei et al. [2025] pretrain a pixel-space diffusion encoder with contrastive semantic learning and representation consistency across noise levels, and then reuses the encoder for downstream generation. Instead of defining positive pairs through hand-crafted image augmentations, MPG uses masked reconstruction to learn visual structure and same-trajectory consistency to adapt the representation to diffusion noise levels.

Masked image modeling for transferable representations. Masked image modeling has become an effective self-supervised paradigm for vision representation learning [He et al., 2022, Hu et al., 2026]. Its generative variants have also been widely explored, including iterative masked-token prediction in MaskGIT [Chang et al., 2022], unified representation learning and synthesis in MAGE [Li et al., 2023], text-to-image generation in Muse [Chang et al., 2023], and masked Transformer diffusion in MaskDiT [Zheng et al., 2023]. Similarly, USP [Chu et al., 2025] employs masked latent modeling to initialize diffusion models for both image generation and visual understanding, whereas our work focuses on pixel-space pretraining for cross-dataset generative transfer. Among them, MaskDiT is particularly related to our work because it introduces high-ratio patch masking into diffusion Transformer training. By using an asymmetric encoder–decoder architecture, MaskDiT lets the encoder process only visible noisy patches and employs a lightweight decoder to recover the full patch sequence, thereby reducing the computational cost of diffusion training. It further combines the standard denoising objective with an auxiliary masked-patch reconstruction task, showing that masking can be beneficial not only for representation learning but also for efficient diffusion model optimization. Despite this architectural similarity, MPG differs from MaskDiT in both objective and usage. MaskDiT applies masking within source-domain diffusion training, with the primary goal

of accelerating and improving the training efficiency of generative models. MPG uses masking in a separate self-supervised pretraining stage to learn a reusable encoder before generative fine-tuning.

3 Preliminaries

Pixel-space diffusion. We follow the EDM-style variance-exploding parameterization [Karras et al., 2022]. Given a clean image $x_0 \sim p_{\text{data}}$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, a noisy sample at noise level t is constructed as

$$x_t = x_0 + t\epsilon, \qquad t \in [\sigma_{\min}, \sigma_{\max}]. \tag{1}$$

A pixel-space denoiser is trained to predict the clean image from the noisy input, typically by minimizing

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0,\epsilon,t}\left[\lambda(t)\,\|D_\theta(x_t, t) - x_0\|_2^2\right], \tag{2}$$

where $\lambda(t)$ is a noise-dependent weighting function. Unlike latent diffusion models that operate after external image compression, pixel-space diffusion directly models images in RGB space, avoiding tokenizer-induced reconstruction errors but making optimization more computationally demanding.

Representation consistency training. In representation consistency training [Lei et al., 2025], the adjacent noisy samples that share the same clean image of different noise levels are approximated [Song et al., 2023] via:

$$x_{t_n} = x_0 + t_n\epsilon, \qquad x_{t_{n-1}} = x_0 + t_{n-1}\epsilon. \tag{3}$$

Let $\{t_i\}_{i=0}^{K-1}$ denote the sequence of discrete noise levels used during pretraining, where K represents the total number of noise scales and $t_0 = \sigma_{\min}$ is the minimum noise level. For any adjacent pair where $t_{n-1} < t_n$, the two samples differ only in noise magnitude and share the same noise sampled from the standard Gaussian distribution. This shared-noise coupling provides a teacher-free way to impose consistency between predictions or representations at adjacent noise levels.

Subsequently, the pretraining objective is formulated as:

$$\mathcal{L}_{\text{rc}} = \mathbb{E}\left[d\left(\Phi_\theta(x_{t_n}, t_n), \Phi_{\text{sg}(\theta)}(x_{t_{n-1}}, t_{n-1})\right)\right], \tag{4}$$

where Φ_θ denotes the noise-conditioned encoder representation, $d(\cdot,\cdot)$ is a representation distance, and $\text{sg}(\cdot)$ denotes stop-gradient. This paradigm shows that Stage-I representation learning can provide a useful initialization for pixel-space generation. Our work keeps the same pretrain–finetune setting, but replaces augmentation-based contrastive semantic learning with masked reconstruction and applies same-trajectory consistency to clean predictions on masked regions.

4 Method

We formulate our diffusion model from the SDE perspective and adopt the notations in the preliminaries (EDM-style noise parameterization). Our overall pipeline contains two stages: (i) self-supervised pretraining, and (ii) generative fine-tuning, as shown in Figure 2.

4.1 Stage I: Self-supervised Pretraining

Architecture. The pretraining network is an asymmetric ViT [Dosovitskiy et al., 2021] composed of an encoder E_θ and a lightweight decoder L_θ. Given a clean image $x_0 \in \mathbb{R}^{3\times H\times W}$, we divide it into non-overlapping $p \times p$ patches and obtain $N = HW/p^2$ patch tokens. We construct noisy inputs as in Eq. (3). During Stage-I pretraining, a random subset of ρN patch tokens is masked, where ρ is the mask ratio. The encoder receives only the visible $(1-\rho)N$ image tokens, together with a learnable `[CLS]` token and a noise-level token. In the clean reconstruction branch, the input is $x_{t_0} = x_0 + t_0\epsilon$ with $t_0 = \sigma_{\min}$. Since t_0 is the minimum noise level, x_{t_0} remains close to the clean image, while keeping the input interface consistent with the diffusion parameterization. The lightweight decoder is trained to predict the clean target patches of x_0. Specifically, the decoder receives the encoded visible tokens and learnable mask tokens inserted at the masked positions, and predicts the corresponding clean image patches. The role of masking in MPG is to define a nontrivial self-supervised reconstruction task. A high mask ratio forces the encoder to infer missing content from broader context, thereby strengthening the visual representations prior to generative fine-tuning.

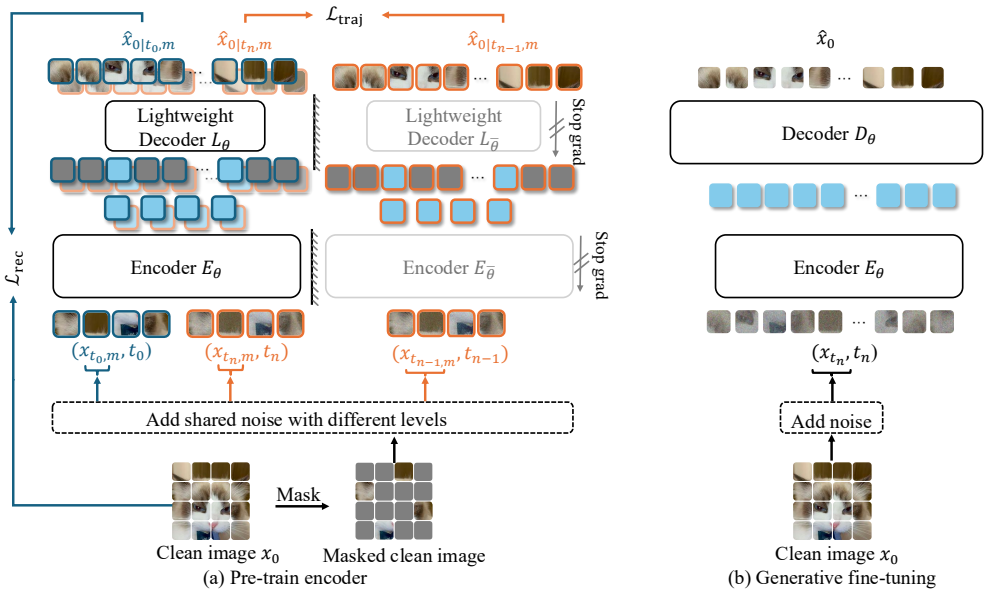

Figure 2: **Method overview.** (a) During pretraining, we learn semantic representations by reconstructing masked regions of the clean image with a masked-reconstruction loss $\mathcal{L}_{\text{rec}}$, and further align the reconstructed outputs along the same ODE sampling trajectories with a masked-consistency loss $\mathcal{L}_{\text{traj}}$. sg denotes the stop-gradient operation. (b) During generative fine-tuning, we discard the pretrained decoder and introduce a randomly initialized decoder that is depth-symmetric to the encoder but wider (i.e., with a larger latent dimension). We train the model end-to-end for diffusion generation.

Pretraining objective. MPG combines two complementary objectives. For notational clarity, we denote the clean prediction produced by the pretraining encoder–decoder pair as

$$\hat{x}_0(t, m; \theta) = L_\theta(E_\theta(x_t, t; m)), \tag{5}$$

where m denotes the random mask pattern. The first objective is a masked clean reconstruction loss at the smallest noise level $t_0 = \sigma_{\min}$:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x_0,\epsilon,m}\left[\|M \odot (\hat{x}_0(t_0, m; \theta) - x_0)\|_2^2\right], \tag{6}$$

where M denotes the binary mask over patch positions and the loss is evaluated only on masked patches. This objective anchors the encoder to clean-image semantic reconstruction while preserving the same noisy-input interface as the downstream diffusion model.

The second objective adapts the pretrained representation to diffusion trajectories. The model is trained to produce consistent clean predictions from these two neighboring noise levels:

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{x_0,\epsilon,t_n,m}\left[\left\|M \odot \left(\hat{x}_0(t_n, m; \theta) - \text{sg}\left[\hat{x}_0(t_{n-1}, m; \bar{\theta})\right]\right)\right\|_2^2\right], \tag{7}$$

where $\bar{\theta}$ denotes the stop-gradient parameters. The full Stage-I objective is

$$\mathcal{L}_{\text{pre}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}}. \tag{8}$$

4.2 Stage II: Generative Fine-tuning

After pretraining, we discard the lightweight decoder L_θ and initialize a new, higher-capacity decoder D_θ. We then fine-tune the encoder–decoder pair (E_θ, D_θ) end-to-end for conditional generation. Following EPG [Lei et al., 2025], we introduce residual connections between the encoder and decoder, and inject conditioning into the decoder via AdaLN modulation. To support class-conditional denoising, we prepend an additional zero embedding to the encoder's original positional embeddings. Moreover, we concatenate the class embedding and the noise-level embedding with the noisy image

tokens, and feed the resulting sequence into the model as input tokens. We emphasize that patch masking is disabled in the generative fine-tuning stage. We train under the EDM paradigm using the denoising objective

$$\arg\min_{\theta} \mathbb{E}\Big[\lambda(t)\big\|D_\theta(E_\theta(x_t,t),y)-x_0\big\|^2\Big] \tag{9}$$

where y denotes the class label and $\lambda(t)$ is the time-dependent weighting function.

In summary, we adopt a two-stage design to decouple representation learning from high-capacity generative modeling. In Stage I, the lightweight decoder prevents the pretraining objective from being dominated by decoder capacity and makes the encoder responsible for reconstructive and trajectory-level structure. In Stage II, the lightweight decoder is discarded and a stronger decoder is introduced for generation. This separation also enables a controlled comparison with EPG-style pretrain–finetune pipelines, where the main variable is the Stage-I self-supervised objective.

4.3 Representation Analysis for Generative Transfer

To better understand the transferability of learned representations, we analyze their sensitivity to hand-crafted augmentations and geometric changes during downstream fine-tuning. We adopt Centered Kernel Nearest-Neighbor Alignment (CKNNA) [Huh et al., 2024, Kornblith et al., 2019], a metric that compares local neighborhood structures across representation spaces. CKNNA has also been employed in diffusion-based generative modeling to assess representation alignment between diffusion model features and pretrained visual representations, as demonstrated in REPA [Yu et al., 2025a] and REPA-E [Leng et al., 2025].

Patch-level CKNNA. Given two feature matrices U and V extracted from the same B images, CKNNA measures the alignment of their local similarity structures. Let $\widetilde{K}_U$ and $\widetilde{K}_V$ denote their centered kernel matrices, and let A_{ij} indicate whether sample j belongs to the k-nearest neighbors of sample i in both feature spaces, excluding self-neighbors. We define

$$\mathrm{CKNNA}_k(U,V)=\frac{\langle A\odot\widetilde{K}_U,\,A\odot\widetilde{K}_V\rangle_F}{\|A\odot\widetilde{K}_U\|_F\|A\odot\widetilde{K}_V\|_F}, \tag{10}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product. A higher score indicates stronger alignment of local representation geometry. We compute CKNNA independently at each spatial patch position to avoid mixing features from unrelated locations. Let $Z_s(\mathcal{X})$ denote the encoder features of an evaluation image set $\mathcal{X}$ at patch position s, with P spatial positions in total.

Augmentation invariance. We first examine how the pretrained representations respond to hand-crafted augmentations. Given an augmentation $\mathcal{T}$, we compare features extracted from the original and augmented views of the same images:

$$\mathcal{A}_{\mathrm{aug}}(\mathcal{T})=\frac{1}{P}\sum_{s=1}^{P}\mathrm{CKNNA}_k\left(Z_s(\mathcal{X}),Z_s(\mathcal{T}(\mathcal{X}))\right). \tag{11}$$

Higher alignment indicates stronger preservation of local neighborhood geometry under the augmentation, reflecting greater augmentation invariance at the representation level.

Pretrain-to-finetune representation drift. We further investigate how pretrained representations change during downstream generative fine-tuning. For a target dataset d, we extract patch features from the same evaluation images using the encoder before and after Stage-II fine-tuning, denoted by Z_s^{pre} and Z_s^{ft}, respectively. We define the representation drift as

$$\mathcal{D}_d=1-\frac{1}{P}\sum_{s=1}^{P}\mathrm{CKNNA}_k\left(Z_s^{\mathrm{pre}}(\mathcal{X}_d),Z_s^{\mathrm{ft}}(\mathcal{X}_d)\right). \tag{12}$$

A lower drift indicates that the local neighborhood geometry learned during pretraining is better preserved after adaptation to the target distribution.

Table 1: Ablation on the lightweight decoder depth. Default settings are highlighted in grey.

LW-dec depth	1	2	4	8
FID	44.78	41.20	39.56	35.07

Noise consistency during pretraining. As a complementary diagnostic, we introduce Noise-Consistency CKNNA (NC-CKNNA) to evaluate representation consistency across adjacent noise levels during pretraining. Unlike the patch-level analyses above, NC-CKNNA measures one-sided neighborhood alignment between encoder `[CLS]` features, using the lower-noise representations as the reference. We aggregate these alignment scores across noise levels and analyze them alongside linear probing accuracy to characterize pretraining dynamics. The detailed formulation is provided in Appendix C.

5 Experiments

We conduct a series of experiments to validate our pretraining model's scalability and transferability, analyze the link between representation geometry and generation performance.

5.1 Experimental Setup

Implementation details of the pretraining model. We adopt a ViT-Base encoder [Dosovitskiy et al., 2021] trained from scratch.

We choose the lightweight decoder depth based on an ablation study conducted with 390k pretraining steps on ImageNet-256 followed by 100k generative fine-tuning steps, as shown in Table 1. Based on these results, we adopt an 8-block Transformer decoder with a hidden dimension of 512, resulting in a total of 111M parameters for the pretraining model. We pretrain MPG on ImageNet at 256×256 resolution [Deng et al., 2009]. In the first-stage masked modeling on ImageNet, we use a masking ratio of 75%. The masked formulation enables stable training with large batch sizes; accordingly, we use a batch size of 4096 and train for 600k steps. We use a relatively large reconstruction weight, $\lambda_{\text{rec}} = 616$, to ensure that the early-stage representation is first anchored to the clean semantic reconstruction objective at the smallest noise level. It is chosen according to the discretization scale of the first noise interval where semantic alignment begins. Empirically, this weighting prevents the trajectory-consistency objective from dominating before the encoder has learned a stable clean-image semantic representation, especially during the early phase of pretraining. We use a time-dependent consistency weight $\lambda_{\text{traj}}(t) = \frac{1}{t_n - t_{n-1}}$.

Implementation details of the generative decoder. Following the asymmetric design in EPG [Lei et al., 2025], we widen the decoder in the generative model. We keep the decoder depth symmetric to the encoder to facilitate residual connections, while increasing the latent dimension from 768 to 1920 and setting `num_heads` $= 16$. This brings the total parameter count of the generative model to 789M. During generative fine-tuning, we disable masked modeling; therefore, we train with a smaller batch size (compared with the pre-train model) of 1024. For transfer, analysis, and ablation experiments, the denoising model is trained for 100k steps. Conversely, source-domain fine-tuning on ImageNet-256 [Deng et al., 2009] is conducted for 800 epochs to align with standard practices in pixel-space diffusion.

Evaluation. We evaluate conditional generation on ImageNet-256. We report standard sample-quality and diversity metrics, including FID [Heusel et al., 2017], Inception Score (IS) [Salimans et al., 2016], Precision/Recall [Kynkäänniemi et al., 2019], together with sampling cost measured by the number of function evaluations (NFE). For transfer, scaling and ablation studies, we utilize the Euler deterministic sampler with 50 steps and report FID calculated on 10K generated samples without classifier-free guidance (CFG) [Ho and Salimans, 2021] to ensure a fair comparison of representation scaling laws. For results on ImageNet-256, we adopt Interval-CFG [Kynkäänniemi et al., 2024] where we use the recommended guidance interval $\sigma \in (0.19, 1.61]$.

Table 2: **Cross-dataset transfer from ImageNet-pretrained encoders.** MPG and EPG use the same encoder backbone and the same Stage-II finetuning protocol. Lower FID is better.

Dataset	EPG	MPG	Rel. Improv.	Setting
CelebA-HQ [Karras et al., 2017]	34.93	**32.48**	**+7.0%**	Uncond.
CIFAR-10 [Krizhevsky et al., 2009]	27.70	**24.70**	**+10.8%**	Cond.
Flowers-102 [Nilsback and Zisserman, 2008]	103.71	**89.70**	**+13.5%**	Cond.
LSUN-Bedroom [Yu et al., 2015]	6.98	**6.11**	**+12.5%**	Uncond.
LSUN-Church [Yu et al., 2015]	34.97	**26.26**	**+24.9%**	Uncond.
Places [Zhou et al., 2014]	23.42	**17.34**	**+26.0%**	Cond.
ADE20K [Zhou et al., 2019]	93.16	**79.24**	**+14.9%**	Uncond.
Average	46.41	**39.40**	**+15.1%**	–

Table 3: **System-level comparison on ImageNet-256 with CFG.**

Model	FID↓	IS↑	Precision↑	Recall↑	Epochs	#Params
ADM [Dhariwal and Nichol, 2021]	3.94	215.8	0.83	0.53	963	673M
FARMER [Zheng et al., 2025]	3.60	269.2	0.81	0.51	320	1.9B
RIN [Jabri et al., 2023]	3.42	182.0	-	-	480	410M
SiD [Hoogeboom et al., 2023]	2.44	256.3	-	-	800	2.46B
MaskDiT [Zheng et al., 2023]	2.28	276.5	0.80	0.61	1600	675M
PixNerd-XL/16(Euler-50) [Wang et al., 2025]	2.29	303.0	0.80	0.59	160	700M
JiT-H/16 [Li and He, 2025]	1.86	303.4	0.78	0.62	600	953M
EPG-XXL/16 [Lei et al., 2025]	1.87	287.6	0.80	0.63	800	789M
MPG	2.21	236.1	0.81	0.59	800	789M

5.2 Main Results: Cross-Dataset Generative Transfer

We first evaluate the central question of this work: whether a pixel-space pretrained encoder can be transferred to generative datasets whose visual statistics differ from the source domain. We consider seven target datasets covering low-resolution natural images, fine-grained categories, faces, indoor/outdoor scenes, and complex layouts. These datasets serve as controlled stress tests for representation transfer under distribution shift. Notably, since several target datasets do not provide categorical labels, we evaluate all models without classifier-free guidance (CFG), reporting the direct diffusion sampling results to ensure a strict and fair comparison of the learned representations. For a fair comparison, MPG and EPG use the same ViT-B encoder backbone and the same Stage-II finetuning protocol. The only intended difference is the Stage-I self-supervised objective: contrastive augmentation-based pretraining for EPG versus masked reconstruction and trajectory consistency for MPG.

Table 2 reports the main cross-dataset transfer results. MPG improves FID on all seven targets, with particularly large gains on fine-grained or scene-centric datasets. To provide a visual intuition for these improvements, qualitative comparisons of the generated samples are presented in Figure 1. These results indicate that removing hand-crafted augmentation design and preserving reconstruction-oriented spatial cues can improve transfer robustness in pixel-space diffusion. Furthermore, to explicitly isolate the contributions of our design choices, we provide a focused component ablation in Appendix Table 8. This analysis confirms that both the reconstruction-oriented masked modeling (predicting clean images) and the trajectory-level representation alignment are crucial drivers for the observed transfer gains.

5.3 Source-Domain ImageNet-256 Generation

We next evaluate MPG on ImageNet-256 to verify its compatibility with large-scale source-domain generation. As shown in Table 3, with an FID of 2.21, MPG demonstrates competitive performance among pixel-space models and slightly outperforms the latent-space competitor MaskDiT and pixel-space competitor PixNerd. While there remains a gap compared to the absolute state-of-the-art

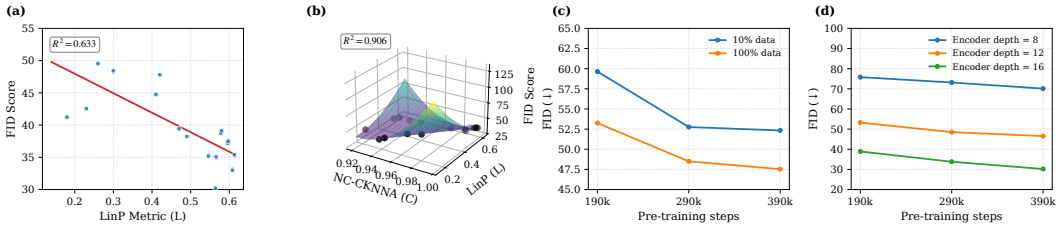

Figure 3: Pretraining diagnostics and controlled scaling within the ImageNet-256 regime.

Table 4: Augmentation sensitivity of pretrained patch representations on Flowers-102, measured using patch-level CKNNA.

Augmentation	Property probed	EPG	MPG	Rel. drop (%)
Color jitter (weak)	Color	0.8159	0.7690	5.75
Color jitter (medium)	Color	0.7002	0.6000	14.31
Color jitter (strong)	Color	0.5268	0.3524	33.11
Gaussian blur ($\sigma = 1.0$)	Texture	0.7871	0.7492	4.82
Grayscale ($p = 0.2$)	Chromatic information	0.8501	0.8027	5.58
Solarize ($p = 0.2$)	Photometric intensity structure	0.8373	0.7736	7.61
Crop (scale $[0.08, 1]$)	Spatial structure	0.3220	0.2958	8.14

Note: Rel. drop $= 100 \times (\text{EPG} - \text{MPG})/\text{EPG}$. Higher CKNNA indicates stronger augmentation invariance.

source-domain generators like JiT-H and EPG-XXL, our work successfully validates the viability of transferable generative pretraining in pixel space. More importantly, it paves the way for future scaling: integrating MAE-style generative pretraining with highly efficient architectures like JiT could potentially yield scalable models with exceptional source-domain generation and downstream transfer abilities.

5.4 Representation Analysis of Generative Transfer

To better understand the transferability of MPG, we conduct complementary analyses of augmentation invariance and representation changes during downstream fine-tuning using patch-level CKNNA, as defined in Section 4.3.

Augmentation invariance. We first examine how pretrained representations respond to hand-crafted augmentations on Flowers-102. For each augmentation, we measure the local representation alignment between original and augmented images using the same spatial patch positions. As shown in Table 4, augmentation-based pretraining consistently exhibits higher CKNNA across all tested transformations. These results indicate that MPG retains greater sensitivity to augmentation-induced visual changes, supporting our motivation to avoid imposing predefined invariances during pretraining.

Representation retention during fine-tuning. We further investigate how pretrained representations evolve during downstream generative fine-tuning. Using identical evaluation images and feature extraction settings, we measure the change in local representation geometry between Stage-I and Stage-II encoders across seven target datasets. We report the mean and standard deviation of patch-geometry drift across repeated evaluation subsets. As shown in Table 5, MPG exhibits lower representation drift than augmentation-based pretraining on all seven datasets. This suggests that MPG better preserves its pretrained local representation geometry during adaptation to different target distributions. These findings are consistent with its stronger cross-dataset generation performance, providing additional evidence for the benefits of augmentation-free pretraining.

Pretraining dynamics. As a complementary analysis, we examine the relationship between representation consistency and downstream generation performance using NC-CKNNA and linear probing (LinP). Figure 3(a) shows that LinP is associated with downstream FID, while Figure 3(b) further considers noise-level representation consistency through NC-CKNNA. A descriptive fit combining

Table 5: Pretrain-to-finetune patch-geometry drift across seven target datasets.

Target dataset	EPG drift ↓	MPG drift ↓
CelebA-HQ [Karras et al., 2017]	0.7233 ± 0.0371	$\mathbf{0.7002 \pm 0.0333}$
CIFAR-10 [Krizhevsky et al., 2009]	0.8050 ± 0.0141	$\mathbf{0.7679 \pm 0.0319}$
Flowers-102 [Nilsback and Zisserman, 2008]	0.8549 ± 0.0339	$\mathbf{0.7396 \pm 0.0391}$
LSUN-Bedroom [Yu et al., 2015]	0.7534 ± 0.0210	$\mathbf{0.6424 \pm 0.0271}$
LSUN-Church [Yu et al., 2015]	0.7874 ± 0.0256	$\mathbf{0.6386 \pm 0.0249}$
Places [Zhou et al., 2014]	0.7539 ± 0.0220	$\mathbf{0.6688 \pm 0.0464}$
ADE20K [Zhou et al., 2019]	0.8210 ± 0.0243	$\mathbf{0.7978 \pm 0.0445}$

both diagnostics achieves $R^2 = 0.906$, compared with $R^2 = 0.633$ for LinP alone on the observed configurations. Although neither metric consistently improves monotonically during pretraining, their joint analysis provides a complementary view of semantic separability and representation consistency across noise levels. We emphasize that this fitted relationship characterizes the observed configurations rather than providing a general predictor of generation quality.

5.5 Controlled Scaling within the ImageNet Pretraining Regime

Finally, we examine whether the proposed Stage-I objective remains effective under controlled increases in pretraining data and encoder capacity. We emphasize that this experiment is not intended to establish a universal scaling law for pixel-space diffusion. Instead, it evaluates a narrower question: within the ImageNet-256 pretraining regime, does masked generative pretraining continue to benefit from more data and a larger encoder when the downstream finetuning protocol is kept fixed?

Figure 3(c,d) summarizes the results. Increasing the pretraining data fraction from ImageNet-10% to the full ImageNet training set improves downstream FID under the same finetuning setup, suggesting that MPG benefits from broader source-domain coverage. Increasing the encoder depth also improves downstream FID at comparable pretraining budgets, indicating that the masked reconstruction and trajectory-consistency objectives remain compatible with larger encoder capacities. These results show that MPG does not saturate under the data and model sizes considered in our experiments. Verifying whether the same trend holds beyond ImageNet-scale pretraining, larger datasets, or different source domains remains an important direction for future work.

6 Discussion and Conclusion

We present MPG, an augmentation-free masked pretraining framework for transferable pixel-space diffusion. By replacing contrastive augmentations with masked reconstruction and trajectory consistency, MPG enables robust encoder transfer across diverse generative datasets. We also introduce NC-CKNNA, a trajectory-level diagnostic metric that complements semantic probes to efficiently evaluate complex pretraining dynamics.

Empirically, MPG presents a trade-off: while it consistently improves cross-dataset transfer performance over EPG-style baselines, its source-domain generation quality on ImageNet-256 does not yet match heavily optimized models like EPG-XXL. However, we argue that transfer robustness — adapting without hand-crafted augmentations—is the primary goal of the pretrain-finetune paradigm. Integrating MPG with specialized architectures like JiT or PixelDiT holds promise for closing this gap, potentially yielding models that excel in both source-domain synthesis and cross-dataset transfer.

Acknowledgments and Disclosure of Funding

Funding. This work was conducted during the first author's internship at AMAP, Alibaba Group, with research support from AMAP, Alibaba Group.

Competing Interests. The authors declare no additional competing interests beyond the affiliations and support disclosed in this paper.

References

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Xiangxiang Chu, Renda Li, and Yong Wang. Usp: Unified self-supervised pretraining for image generation and understanding. *ICCV*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 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. *Advances in Neural Information Processing Systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18062–18071, 2025. doi: 10.1109/CVPR52734.2025.01683.

Yiming Hu, Xiangxiang Chu, and Yong Wang. Robust mae-driven nas: From mask reconstruction to architecture innovation. *arXiv preprint arXiv:2311.12086*, 2026.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Allan Jabri, David J Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *International Conference on Machine Learning*, pages 14569–14589. PMLR, 2023.

Dongyang Jin, Ryan Xu, Jianhao Zeng, Rui Lan, Yancheng Bai, Lei Sun, and Xiangxiang Chu. Semantic context matters: Improving conditioning for autoregressive models. *arXiv preprint arXiv:2511.14063*, 2025.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. *Advances in Neural Information Processing Systems*, 34:17710–17722, 2021.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no vae: End-to-end pixel-space generative modeling via self-supervised pre-training. *arXiv preprint arXiv:2510.12586*, 2025.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025.

Huaqiu Li, Yong Wang, Tongwen Huang, Hailang Huang, Haoqian Wang, and Xiangxiang Chu. Ld-rps: Zero-shot unified image restoration via latent diffusion recurrent posterior sampling. *ICCV*, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2142–2152, 2023.

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*, pages 23–40. Springer, 2024.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. *arXiv preprint arXiv:2511.19365*, 2025.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. *arXiv preprint arXiv:2506.05350*, 2025.

Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. *arXiv preprint arXiv:2411.19722*, 2024.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 2025.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025.

Ryan Xu, Dongyang Jin, Yancheng Bai, Rui Lan, Xu Duan, Lei Sun, and Xiangxiang Chu. Scalar: Scale-wise controllable visual autoregressive learning. *arXiv preprint arXiv:2507.19946*, 2025.

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*, pages 15703–15712, 2025.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Meng Yu, Lei Sun, Jianhao Zeng, Xiangxiang Chu, and Kun Zhan. Elucidating the snr-t bias of diffusion probabilistic models. *arXiv preprint arXiv:2604.16044*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=DJSZGGZYVi.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025b.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. *arXiv preprint arXiv:2510.23588*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

A The Use of Large Language Models (LLMs)

To enhance the clarity and readability of this work, we utilized DeepSeek[3] for language refinement in several sections (e.g., abstract and introduction) of the manuscript. The tool's suggestions were reviewed and edited to ensure technical accuracy and alignment with the paper's scientific content.

B Implementation Details

B.1 Baseline methods

As shown in Table 6, we introduce the implementation details for baseline method used in Table 2 and Table 8. We use the official checkpoint of pre-trained EPG encoder from Lei et al. [2025]. Its encoder shares the same architecture as MPG and is pre-trained with a batch size of 4096 for 600k steps.

Table 6: Network configurations in transfer experiments. Encoder and decoder settings are separated by comma.

Model	Blocks	Dim	Heads	Params
EPG	12, 12	768, 768	12, 12	228M
MPG	12, 12	768, 768	12, 12	228M

Table 7 summarizes the network configurations used for ImageNet-256 source-domain pre-training and generative fine-tuning. The encoder architecture is kept consistent across the two stages to ensure that the representation learned during self-supervised pre-training can be directly transferred to diffusion fine-tuning. In Stage-I, MPG adopts an asymmetric ViT architecture with a lightweight decoder, following the common design principle of masked image modeling: the decoder has limited capacity so that the encoder is encouraged to learn reconstructive and trajectory-level visual representations. After pre-training, the lightweight decoder is discarded. Stage-II initializes the encoder from the pretrained weights and attaches a higher-capacity generative decoder for EDM-style diffusion training. This two-stage configuration allows us to separate the effect of the pre-training objective from the capacity of the final generative model.

B.2 Implementation details

Data pre-processing. We pre-process ImageNet dataset by center-cropping images into 256×256 and save images in lossless PNG format.

Training settings. We adopt the definition of SDEs and the time discretization strategy from EDM [Karras et al., 2022]. The time horizon is divided into N non-overlapping intervals, where each interval endpoint n is mapped to discretized time step (or noise level) via the following equation

$$t_n = (\sigma_{min}^{1/\rho} + \frac{n}{N-1}(\sigma_{max}^{1/\rho} - \sigma_{min}^{1/\rho}))^{\rho}, \quad \text{with} \quad n \in [0, N-1]. \tag{13}$$

Here, $\sigma_{max} = 80, \sigma_{min} = 0.002, \rho = 7$, N is total discretization step.

We consider our pre-training as a combination of representation learning [He et al., 2022] and consistency training [Song and Dhariwal, 2023]. Built on top of these, we adopt the hyperparameters used in our ImageNet-256 experiments, as detailed in 9 and 10.

B.3 Component Ablation

To assess the contributions of masked reconstruction and trajectory consistency, we conduct a controlled ablation on Flowers-102 under a reduced fine-tuning budget of 50K steps. All variants follow the same Stage-II fine-tuning and evaluation protocol, with FID computed against the same held-out test-reference statistics used in the main transfer experiments. Pure masked reconstruction

[3]https://www.deepseek.com

Table 7: **Network configurations used for ImageNet-256 source-domain pre-training and generative fine-tuning.** We report the architecture settings used in the two-stage MPG pipeline. For each stage, encoder and decoder configurations are separated by commas.

Stage	Model	Blocks	Dim	Heads	Params
Stage-I pre-training	MPG-Pretrain	12, 8	768, 512	12, 8	111M
Stage-II fine-tuning	MPG	12, 12	768, 1920	12, 16	789M

Table 8: Component ablation on Flowers-102 after 50K steps of generative fine-tuning.

Stage-I initialization	FID ↓
No pretraining	108.28
EPG	112.44
Pure masked reconstruction	104.76
MPG	**97.83**

adopts an MAE-style objective on clean images, while MPG additionally enforces consistency between masked clean-image predictions at adjacent noise levels.

As shown in Table 8, pure masked reconstruction improves FID over training from scratch (104.76 vs. 108.28), while MPG further reduces FID to 97.83, outperforming both pure masked reconstruction and EPG (112.44). These results suggest that trajectory consistency provides additional benefits beyond masked reconstruction alone. The MPG advantage over EPG is also maintained after 100K fine-tuning steps (89.70 vs. 103.71), consistent with the main transfer results.

B.4 Broader impacts.

This work studies reusable pixel-space diffusion pretraining. Its potential positive impact lies in reducing the cost of training generative models on downstream datasets and improving transfer robustness. However, improvements in image generation and transferability may also be misused to produce misleading synthetic images or other harmful visual content. Since this paper does not release a pretrained generator or a new scraped dataset, the immediate release risk is limited; if model checkpoints are released in the future, appropriate usage guidelines and release safeguards should be considered.

Table 9: Our pre-training and fine-tuning diffusion settings. †: In denoising training, we first sample t from a continuous lognormal distribution and then map it to the nearest discrete value t_n. ‡: the denoiser predicts clean image $\mathbf{x}$ at different noise levels.

	Pre-training	Denoising Training
Total discretization steps N	from 20 to 1280	1280
Weighting $\lambda(t)$	$\frac{1}{t_n - t_{n-1}}$	$\frac{1}{t_n - t_{n-1}}$
Preconditioning	$1/\sqrt{t_n^2 + \sigma_{data}^2}$	$1/\sqrt{t_n^2 + \sigma_{data}^2}$
Time condition	$c(t) = 1000 * \frac{1}{4} \ln t$	$c(t) = 1000 * \frac{1}{4} \ln t$
Noise Sampling	$t_n \sim \text{lognormal}(-0.4, 1.6)^\dagger$	$t_n \sim \text{lognormal}(-1.2, 1.6)^\dagger$
Training objective	Eq. (8)	Eq. (9)
Model parameterization	-	$\mathbf{x}$-prediction‡
Shifting of noise level	image size/64	image size/64

C Details of NC-CKNNA

This appendix provides the formulation and implementation details of Noise-Consistency CKNNA (NC-CKNNA), a complementary diagnostic for analyzing representation consistency across noise levels during pretraining. Building on CKNNA-style local kernel alignment [Huh et al., 2024, Leng et al., 2025], we adopt a one-sided neighborhood mask to measure how local representation geometry is preserved across adjacent noise levels.

Table 10: Hyper-parameters.

	Pretraining	Fine-tuning
Bs	4096	1024
Training steps	600k	1000k
Lr	6e-4, Cosine Decay	1e-4
Optimizer	AdamW(0.90, 0.95)	AdamW(0.9, 0.999)
Wd	0.05	0.05

C.1 Noise-conditioned features and reference neighborhoods

Given a batch of B clean images $\{x_i\}_{i=1}^{B}$, we construct noisy samples at discrete noise levels $\{t_n\}_{n=0}^{K-1}$:

$$x_{t_n}^{(i)} = x_i + t_n \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I), \tag{14}$$

where the same noise realization ϵ_i is shared across noise levels for each image.

We extract the encoder's `[CLS]` features and apply ℓ_2 normalization to obtain $r_n(x_i) \in \mathbb{R}^D$. The feature matrix and its corresponding kernel matrices are defined as

$$\begin{aligned} R_n &= [r_n(x_1), \ldots, r_n(x_B)]^\top, \\ K_n &= R_n R_n^\top, \\ \widetilde{K}_n &= H K_n H, \quad H = I - \frac{1}{B}\mathbf{1}\mathbf{1}^\top. \end{aligned} \tag{15}$$

For each anchor i, we identify its k nearest neighbors $\mathcal{N}_k^{(n-1)}(i)$ using the uncentered cosine similarities K_{n-1}, excluding self-neighbors. We then construct a one-sided reference mask:

$$A_{ij}^{(n-1)} = \mathbb{I}\left[j \in \mathcal{N}_k^{(n-1)}(i)\right]. \tag{16}$$

This mask fixes the neighborhoods defined by the lower-noise representations, allowing us to examine their preservation at the next higher noise level.

C.2 One-sided neighborhood alignment

Using the same reference mask for both centered kernel matrices, we define

$$\mathrm{NCCKNNA}_n = \frac{\left\langle \widetilde{K}_{n-1} \odot A^{(n-1)}, \widetilde{K}_n \odot A^{(n-1)} \right\rangle_F}{\left\| \widetilde{K}_{n-1} \odot A^{(n-1)} \right\|_F \left\| \widetilde{K}_n \odot A^{(n-1)} \right\|_F}. \tag{17}$$

Here, $\odot$ denotes element-wise multiplication and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product. Unlike standard CKNNA, which considers neighborhoods from both representation spaces, NC-CKNNA fixes the lower-noise neighborhoods as a reference. A higher score indicates better preservation of local similarity structures across adjacent noise levels.

C.3 Aggregation and implementation details

We aggregate the alignment scores across noise levels using a weighted average:

$$\mathrm{NCCKNNA} = \frac{\sum_{n=1}^{K-1} \lambda_n \mathrm{NCCKNNA}_n}{\sum_{n=1}^{K-1} \lambda_n}, \qquad \lambda_n = \frac{1}{t_n - t_{n-1}}. \tag{18}$$

Following the inverse step-size weighting in iCT [Song and Dhariwal, 2023], this aggregation assigns greater weight to shorter noise-level intervals.

Unless otherwise stated, we use $B = 1000$ evaluation images and $k = 10$ nearest neighbors. Features are ℓ_2-normalized before kernel construction. Self-neighbors are excluded, ties are broken deterministically, and a small numerical constant is added to the normalization denominator when necessary.

D Joint Analysis of Noise Consistency and Semantic Separability

To further examine the relationship between pretraining representations and downstream generation quality, we analyze the interaction between noise-level consistency and semantic separability. Let C denote the aggregated NC-CKNNA score and L denote linear probing (LinP) accuracy. We use an empirical joint model to summarize their relationship with downstream FID:

$$\widehat{\mathrm{FID}}(C, L) = \phi(\alpha C + \beta L) + \lambda \left[L - (kC + b)\right]^2, \tag{19}$$

where $\phi(\cdot)$ is a low-degree polynomial, α and β weight the two diagnostics, and $\lambda \geq 0$ controls the contribution of their deviation from an empirically fitted relation $L = kC + b$. The parameters are estimated from observed (C, L, FID) triplets by minimizing the fitting error.

As illustrated in Figure 3(b), the joint fit achieves an in-sample R^2 of 0.906, compared with 0.633 for the LinP-only fit. This suggests that considering both semantic separability and noise-level consistency provides a more complete description of the observed pretraining configurations. However, the fitted surface is intended solely as a descriptive analysis rather than a general predictor of downstream generation quality.

E Datasets for Cross-Dataset Transfer

This appendix summarizes the seven downstream datasets used for cross-dataset generative transfer from an ImageNet-256 pre-trained checkpoint. Our goal is to test *transfer robustness* under domain shift and dataset-scale variation, covering small fine-grained benchmarks (e.g., Flowers), human-centric imagery (CelebA-HQ), scene-centric distributions (Places), and complex structured layouts (ADE20K), as well as LSUN subsets with distinct spatial statistics (church/bedroom). These targets differ substantially from ImageNet in texture, object granularity, background complexity, and semantic organization, making them a useful stress-test for pretrain–finetune workflows in pixel-space diffusion.

Why these datasets for generative transfer. Compared with ImageNet-256, these datasets exhibit stronger distribution shift in at least one of the following aspects: (i) *semantic granularity* (Flowers-102 contains fine-grained categories with subtle inter-class cues), (ii) *domain specialization* (CelebA-HQ is face-centric and departs from ImageNet object-centric composition), (iii) *global layout and scene structure* (Places and ADE20K emphasize scene-level organization and spatial arrangement), and (iv) *dataset scale and diversity* (some targets are substantially smaller or more homogeneous). Such shifts are precisely where EPG-style pretraining can be promising but also brittle.

In particular, contrastive pretraining pipelines often rely on handcrafted, dataset-agnostic augmentations to enforce invariances. Under distribution shift, these invariances can become misaligned with what downstream generative modeling requires: aggressive crops, color jitter, or distortions may remove or alter the dataset-specific cues that are crucial for faithful generation, especially on small or fine-grained targets. Moreover, the invariances induced by contrastive objectives can oversuppress nuisance variations differently across domains, leading to unstable finetuning behavior when the downstream dataset statistics depart from ImageNet. Our evaluation therefore focuses on whether augmentation-free, reconstruction-oriented pretraining improves transfer stability, and whether alignment/semantics signals can predict downstream behavior.

Data sources. CIFAR-10 is loaded via the standard `torchvision` dataset interface. CelebA-HQ (256) is obtained from a Kaggle release. All remaining datasets are downloaded from the Hugging Face (HF) datasets hub. For reproducibility, we list the exact dataset identifiers used in our experiments: ADE20K: `sezer12138/ade20k_`; LSUN-Bedroom: `pcuenq/lsun-bedrooms`; Flowers-102: `nkirschi/oxford-flowers`; Places: `ljnlonoljpiljm/places365-256px`; LSUN-Church: `tglcourse/lsun_church_train`. Unless otherwise stated, we use the full training split provided by these entries.

Class labels and conditional transfer. All datasets listed below provide labels or metadata in their standard releases, but we *only* use class-conditional generation on **CIFAR-10**, **Places**, and **Flowers-102**. The remaining targets are fine-tuned and evaluated in an unconditional generation setting.

Table 11: Downstream datasets for cross-dataset transfer.

Dataset	Domain	Labels in release
CIFAR-10 [Krizhevsky et al., 2009]	natural images (objects)	yes (10 classes)
Flowers-102 [Nilsback and Zisserman, 2008]	fine-grained flowers	yes (102 classes)
CelebA-HQ 256 [Karras et al., 2017]	faces (high quality)	attributes/IDs
Places [Zhou et al., 2014]	scenes	yes (365 classes)
ADE20K [Zhou et al., 2019]	complex scenes/layouts	semantic labels
LSUN-Church [Yu et al., 2015]	church exteriors	subset labels
LSUN-Bedroom [Yu et al., 2015]	indoor bedrooms	IDs

Per-dataset notes. **CIFAR-10** [Krizhevsky et al., 2009] contains small natural images across 10 object categories. It is commonly used to test transfer from large-scale pretraining to low-resolution targets. Its statistics (resolution, object scale, and backgrounds) differ from ImageNet-256, making it a useful probe for whether pretraining learns transferable structure beyond augmentation-induced invariances. We use class-conditional finetuning.

Flowers-102 [Nilsback and Zisserman, 2008] is a fine-grained dataset where category differences often rely on subtle texture and shape cues. This setting is especially sensitive to augmentation choices: invariances encouraged by contrastive objectives can suppress fine-grained details and lead to unstable generative finetuning on smaller targets. We use class-conditional finetuning.

CelebA-HQ 256 [Karras et al., 2017] is a face-centric, high-quality dataset with strong domain specialization (portraits, aligned facial structure, consistent semantics). This distribution shift from ImageNet is substantial; stable transfer requires preserving structural cues rather than enforcing generic invariances. We fine-tune unconditionally.

Places [Zhou et al., 2014] emphasizes scene semantics and global layout, differing from ImageNet's object-centric bias. This target tests whether pretraining captures spatial organization that benefits generation under scene-level distribution shift. We use class-conditional finetuning.

ADE20K [Zhou et al., 2019] contains complex scenes with diverse layouts and cluttered compositions. Although the dataset is often used for semantic segmentation, here it serves as a challenging generative target where spatial structure and scene composition matter. We fine-tune unconditionally.

LSUN-Church and **LSUN-Bedroom** [Yu et al., 2015] are large-scale scene datasets with distinct spatial statistics (outdoor architecture vs. indoor room structure). They probe transfer across scene types and test whether pretraining preserves geometry- and layout-related cues. We fine-tune unconditionally.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that the paper proposes MPG, an augmentation-free masked generative pretraining framework for transferable pixel-space diffusion, and introduces NC-CKNNA as a trajectory-level diagnostic metric. The claims are supported by the cross-dataset transfer experiments, component ablations, ImageNet-256 evaluation, and controlled scaling analyses reported in Secs. 5.2–5.5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly clarifies that the ImageNet-256 result is intended as source-domain validation rather than state-of-the-art superiority, and that the scaling analysis is restricted to the ImageNet-256 pretraining regime. The conclusion further notes that MPG does not yet surpass the strongest source-domain generators using additional feature-alignment losses or specialized architectures.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical formulations define the diffusion setting, pretraining objectives, and the NC-CKNNA diagnostic metric.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method architecture, pretraining objective, fine-tuning objective, datasets, model sizes, batch sizes, training steps, sampling protocol, evaluation metrics, and ablation settings are described in Secs. 4 and 5.1. These details provide the main information needed to reproduce the reported experimental comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments are conducted on publicly available datasets that are cited in the paper, but the current submission does not include an anonymized code release or full reproduction scripts. We plan to release the code and training/evaluation scripts upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5.1 specifies the encoder and decoder configurations, parameter counts, masking ratio, pretraining steps, batch sizes, fine-tuning budget, evaluation metrics, sampler, number of generated samples, NFE, and CFG setting. The ablation and transfer settings are also described alongside the corresponding tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper does not report error bars or repeated-run confidence intervals. Repeating large-scale pixel-space diffusion pretraining and fine-tuning across all datasets is computationally expensive, so we instead provide controlled comparisons under matched architectures and training protocols, together with component ablations and scaling analyses.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Sec. 5.1 reports model sizes, batch sizes, training steps, sampling cost, and evaluation settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research uses standard public image datasets and studies self-supervised pretraining and diffusion generation. We are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix B.4 discusses both potential positive impacts, such as reducing the cost of reusable generative training, and potential negative impacts, including misuse of improved image generation for misleading synthetic content.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The current submission does not release a pretrained image generator or a new scraped dataset. If pretrained checkpoints are released in the future, we will include appropriate usage guidelines and safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the original datasets and baseline methods and provides dataset source identifiers in Appendix E. However, the current version does not fully enumerate the license or terms of use for every dataset, checkpoint, and code asset, so we answer No.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset, benchmark, or released model asset in the current submission. The proposed method and metric are documented in the main paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, user studies, or crowdsourcing experiments, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for language refinement, as disclosed in Appendix A, and were not used as an important, original, or non-standard component of the proposed method, experiments, or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.