ORDERED REPRESENTATIONS FOR PIXEL DIFFUSION: DENOISING WITH LEARNED VELOCITY CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent pixel diffusion models improve generation through clean-image prediction (x-pred) or encoder–decoder architectures that reduce the noise-modeling burden on the backbone. Combining these designs, we find that x-pred learns stronger representations and achieves lower FID than velocity prediction (v-pred) with sufficient sampling steps, yet underperforms in few-step sampling. Examining its induced sampling velocity reveals that the x-to-v conversion amplifies prediction errors, which are most pronounced along the leading, high-energy directions of RGB images. Motivated by this structure, we propose **OiT**, an Ordered-representation image Transformer that uses nested channel masking to make dominant image information progressively accessible from the leading feature channels. A compact prefix of this representation guides a full-dimensional velocity correction, while the full representation remains responsible for clean-image prediction. This allows OiT to improve the representation quality and long-step generation of x-pred while achieving the strong few-step performance of v-pred. We further introduce **bottlenecked Representation Alignment (bREPA)** and a **Hann Decoder** as lightweight designs for representation alignment and patch decoding. On ImageNet 256×256, OiT-B and OiT-XL achieve FID-50K scores of **2.52** and **1.55**, respectively, reaching state-of-the-art performance among pixel diffusion models at comparable compute.

1 INTRODUCTION

Diffusion models have become a leading paradigm for visual generation (Ho et al., 2020; Dhariwal & Nichol, 2021; Esser et al., 2024; Wu et al., 2025). Latent diffusion models predict velocity (v-pred) in a compact space learned by a pretrained autoencoder (Ma et al., 2024; Lipman et al., 2023; Liu et al., 2023). However, lossy compression in the frozen autoencoder can limit visual fidelity (Wang et al., 2026). Recent pixel diffusion models bypass this bottleneck by denoising images directly (Li & He, 2026; Yu et al., 2026; Wang et al., 2026). In this high-dimensional space, noise or velocity prediction must account for both image content and the exact noise pattern corrupting each image, placing a heavier burden on the denoiser (Li & He, 2026; Zheng et al., 2026).

Prior pixel diffusion models address this challenge in two complementary ways. One line of work adopts clean-image prediction (x-pred), leveraging the structure of natural images instead of directly predicting noise or velocity (Li & He, 2026; Shin et al., 2026). Another introduces encoder–decoder architectures that route the noisy input directly to the decoder, reducing the noise-dependent information that must pass through the backbone (Yu et al., 2026; Wang et al., 2026). Combining these designs further improves pixel-space generation, yet an important limitation remains. Compared with v-pred, x-pred learns stronger representations and achieves lower FID with sufficient sampling steps, but underperforms in few-step sampling, as shown by the 10- and 50-step FID results in Tab. 1.

Table 1: **ImageNet probing and generation.** OiT-B-scale models with 80-epoch training and 30-epoch attentive probing.

Method	Acc.↑	FID@10↓	FID@50↓
v-pred	41.78	<u>12.22</u>	10.65
x-pred	<u>44.12</u>	15.13	<u>8.36</u>
OiT	**45.30**	**12.06**	**7.67**

Although x-pred estimates a clean image $\hat{x}$, sampling relies on its induced velocity $\hat{v}_x = (\hat{x} - z_t)/(1-t)$, which amplifies clean-image prediction errors by $1/(1-t)$. To identify where these errors are most pronounced, we perform PCA on RGB images and measure the velocity error along the resulting principal directions. As shown in Fig. 1(a), the largest errors occur along the leading,

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.

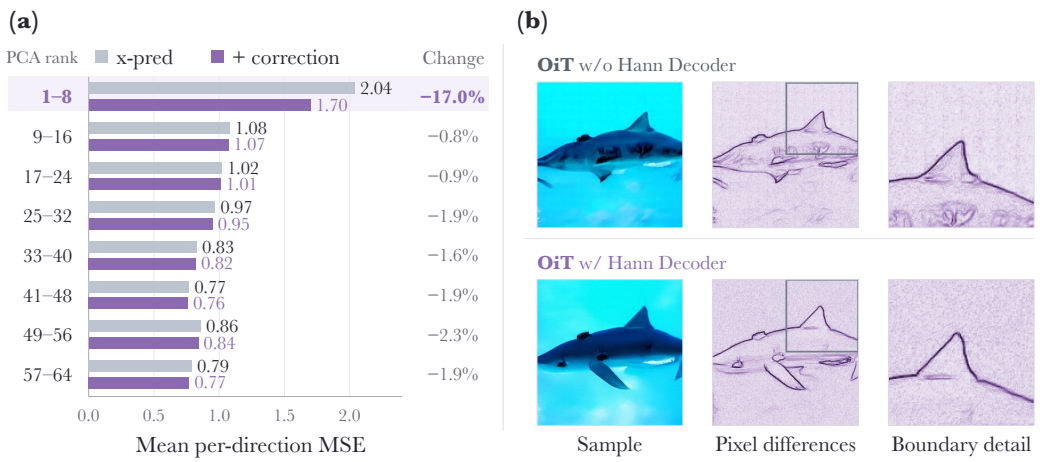

Figure 1: **Velocity correction and Hann decoding. (a) Velocity errors along principal image directions.** We project velocity errors onto PCA directions computed from RGB images and order them by variance. The leading directions exhibit the largest errors and benefit most from our velocity correction. **(b) Patch-boundary artifacts.** Standard patch-wise decoding produces periodic boundary patterns, visible in the pixel-difference maps and enlarged regions. The Hann Decoder blends overlapping predictions and substantially suppresses these artifacts.

high-variance components, suggesting that velocity correction should focus on the dominant image information captured by these directions.

To this end, we propose **OiT**, an **O**rdered-representation **i**mage **T**ransformer that uses nested channel masking (Rippel et al., 2014; Chen et al., 2025b) to structure its representation such that predictive image information is progressively accessible from the leading channels. Rather than correcting in a fixed PCA space (Chen et al., 2026a), OiT uses a learned prefix of this representation to guide a full-dimensional velocity correction, while the full representation remains responsible for clean-image prediction. This design preserves the representation quality and long-step generation of x-pred while matching the strong few-step performance of v-pred. As shown in Tab. 1, OiT improves both representation quality and generation across sampling budgets, while Fig. 1(a) shows that the learned correction substantially reduces the dominant velocity errors.

We further revisit representation alignment and patch decoding in pixel diffusion. Representation alignment is effective in latent diffusion (Yu et al., 2025), while PixelREPA (Shin et al., 2026) identifies an information mismatch when such alignment is applied to pixel representations. Rather than transforming the targets with a Masked Transformer Adapter as in PixelREPA, our **bottlenecked Representation Alignment (bREPA)** simply places a narrow bottleneck in the auxiliary alignment branch, preserving the full generative representation while achieving better performance in our experiments. For patch decoding, conventional patch-wise prediction produces the periodic boundary artifacts shown in Fig. 1(b). Our **Hann Decoder** suppresses them by blending overlapping patch predictions with normalized Hann weights (Harris, 1978; Pielawski & Wählby, 2020).

On ImageNet 256×256, OiT-B achieves an FID of **2.52**, and OiT-XL reaches **1.55**, attaining state-of-the-art performance with comparable per-forward compute among pixel diffusion models. Our main contributions are:

- We introduce **OiT**, which learns ordered representations through nested channel masking and uses a compact feature prefix to guide a full-dimensional velocity correction, combining the representation and long-step strengths of x-pred with strong few-step sampling.
- We introduce two lightweight supporting designs: **bREPA** for representation alignment, which bottlenecks only the auxiliary alignment branch, and the **Hann Decoder** for patch decoding, which suppresses boundary artifacts through overlapping Hann-weighted reconstruction.
- OiT consistently improves both representation quality and generation across sampling budgets, and achieves state-of-the-art class-conditional generation with comparable per-forward compute.

2 PRELIMINARIES

Flow matching. In the linear-path formulation of flow matching (Lipman et al., 2023; Liu et al., 2023), a clean image $x \sim p_{\text{data}}$ and independent Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ are interpolated as

$$z_t = tx + (1-t)\epsilon, \qquad t \in [0,1], \tag{1}$$

with conditional velocity $u = x - \epsilon$. A parameterized velocity field $v_\theta(z_t, t)$ is trained to regress this target by minimizing

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x,\epsilon,t}\left[\|v_\theta(z_t,t) - u\|_2^2\right]. \tag{2}$$

Generation starts from $z_0 \sim \mathcal{N}(0, I)$ and integrates $\mathrm{d}z_t/\mathrm{d}t = v_\theta(z_t, t)$ from $t = 0$ to $t = 1$.

Clean-image prediction. Instead of predicting velocity directly, x-pred estimates the clean image $\hat{x}_\theta(z_t, t)$ (Li & He, 2026) and induces a velocity field through

$$\hat{v}_x(z_t,t) = \frac{\hat{x}_\theta(z_t,t) - z_t}{1-t}, \qquad t < 1. \tag{3}$$

Replacing v_θ with $\hat{v}_x$ in Eq. (2) trains the clean-image predictor through the same velocity-space objective. The network thus predicts image content, while its induced velocity governs sampling.

3 METHOD

OiT retains clean-image prediction and augments its induced velocity with a correction conditioned on ordered features. As illustrated in Fig. 2, the two prediction branches share an encoder and each has direct access to the noisy image. We first introduce nested channel masking for learning ordered representations in Sec. 3.1, then describe how their leading channels guide velocity correction in Sec. 3.2. We further present bREPA for auxiliary semantic supervision in Sec. 3.3 and the Hann Decoder for cross-scale conditioning and overlapping reconstruction in Sec. 3.4.

3.1 ORDERED REPRESENTATIONS

A compact correction branch requires a feature prefix that remains informative without access to later channels. We encourage this property by training the clean decoder to reconstruct images from prefixes of different widths.

Nested channel masking. The encoder processes non-overlapping 16×16 image patches and produces features $h = E_\theta(z_t, t) \in \mathbb{R}^{N \times D}$, where $N = (H/16)(W/16)$ is the number of spatial tokens and D is the channel dimension. Following nested dropout (Rippel et al., 2014), we sample a retained width K for each training example and replace the remaining channels with a learned null representation,

$$\tilde{h}^{(K)} = m^{(K)} \odot h + \left(1 - m^{(K)}\right) \odot h_{\text{null}}, \qquad m_j^{(K)} = \mathbf{1}\{j \le K\}. \tag{4}$$

As shown in Fig. 2(a), the mask acts on channels and is shared across all spatial tokens within an example, leaving the spatial grid intact. We sample K from the full width and log-spaced prefixes with minimum width r. Leading channels participate in more reconstruction tasks and remain predictive when later channels are unavailable. In particular, the first r channels are retained for every example.

Sampled-prefix clean prediction. The clean decoder combines the sampled representation with the noisy image to predict

$$\hat{x}_K = D_x(\tilde{h}^{(K)}, z_t, t). \tag{5}$$

Direct access to z_t allows noise-dependent detail to bypass the encoder, while the feature prefix supplies image-dependent context. The conditioning and reconstruction are described in Sec. 3.4.

The x-pred induces a velocity with a denominator floor for numerical stability (Li & He, 2026):

$$a_K = \frac{\hat{x}_K - z_t}{\max(1-t, t_\epsilon)}, \tag{6}$$

where $t_\epsilon > 0$. The clean branch is trained with

$$\mathcal{L}_{\text{clean}} = \mathbb{E}_{x,\epsilon,t,K}\left[\left\|\frac{x - \hat{x}_K}{\max(1-t,t_\epsilon)}\right\|_2^2\right]. \tag{7}$$

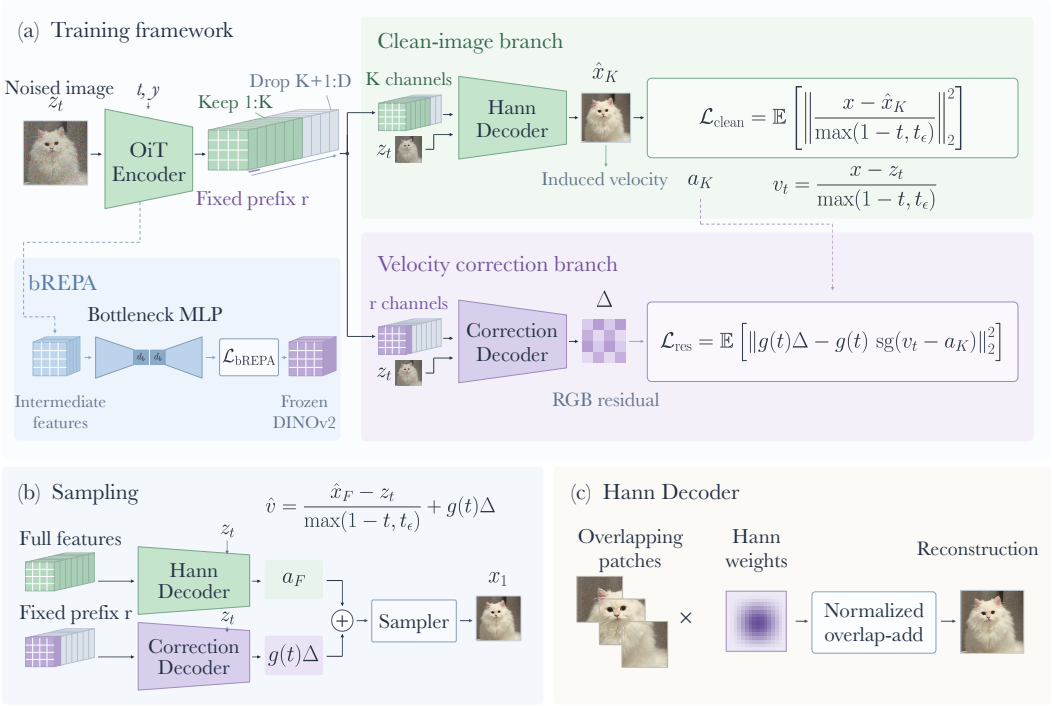

Figure 2: **Overview of OiT. (a) Training framework.** A sampled channel prefix conditions clean-image prediction, while a fixed prefix conditions velocity correction. The residual target uses the clean prediction from the same forward pass. bREPA aligns intermediate encoder features through a narrow auxiliary projection. **(b) Sampling.** Full-width clean-image prediction is combined with gated velocity correction. **(c) Hann Decoder.** Overlapping patch predictions are blended with normalized Hann weights to suppress boundary artifacts.

3.2 Learned Velocity Correction

Clean-to-velocity error. For the unfloored conversion in Eq. (3), clean-image and velocity errors satisfy

$$\hat{v}_x - u = \frac{\hat{x}_\theta - x}{1-t}, \qquad \|\hat{v}_x - u\|_2^2 = \frac{\|\hat{x}_\theta - x\|_2^2}{(1-t)^2}. \tag{8}$$

The conversion magnifies a given reconstruction error by $1/(1-t)$. We therefore retain x-pred and learn an additive residual directly in velocity space, targeting the field used for sampling.

Prefix-conditioned correction. The correction branch in Fig. 2(a) receives the always-retained prefix together with the current noisy image,

$$\Delta = D_v(\tilde{h}^{(r)}, z_t, t) \in \mathbb{R}^{H \times W \times 3}. \tag{9}$$

The first r channels provide image-dependent encoder context, while the suffix is replaced by learned nulls as in Eq. (4). This limits the conditioning information, not the output dimension: Δ is a full RGB velocity residual rather than a correction confined to a prescribed pixel subspace.

The correction decoder has its own Transformer blocks and RGB readout, while sharing input embeddings and conditioning projections with the clean decoder. Both branches use the cross-scale conditioning and Hann synthesis in Sec. 3.4.

Residual supervision. The correction target is the discrepancy between the path velocity $v_t = x - \epsilon$ and the induced velocity a_K from the same sampled-prefix forward pass,

$$v_t - a_K = x - \epsilon - \frac{\hat{x}_K - z_t}{\max(1-t, t_\epsilon)}. \tag{10}$$

The correction branch minimizes

$$\mathcal{L}_{\text{res}} = \mathbb{E}_{x,\epsilon,t,K}\left[\|g(t)\Delta - g(t)\,\text{sg}(v_t - a_K)\|_2^2\right], \tag{11}$$

where $\text{sg}(\cdot)$ stops gradients through the target. The target follows the sampled prediction $\hat{x}_K$, while the correction decoder always receives the fixed prefix of width r.

We use $g(t) = t^2$ to emphasize correction at later timesteps. Residual supervision trains the correction decoder and backpropagates through the semantic prefix, while the target, initial noisy-image state, and global condition are detached.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{clean}} + \lambda_{\text{res}}\mathcal{L}_{\text{res}} + \lambda_{\text{align}}\mathcal{L}_{\text{bREPA}}, \tag{12}$$

where $\mathcal{L}_{\text{bREPA}}$ is the auxiliary alignment loss defined in Sec. 3.3.

Sampling. At inference, the clean decoder uses the full representation, while the correction decoder retains its fixed prefix, as shown in Fig. 2(b). With $\hat{x}_F = D_x(h, z_t, t)$, the sampling field becomes

$$\hat{v} = \frac{\hat{x}_F - z_t}{\max(1-t, t_\epsilon)} + g(t)D_v(\tilde{h}^{(r)}, z_t, t). \tag{13}$$

Starting from Gaussian noise, we integrate $\mathrm{d}z_t/\mathrm{d}t = \hat{v}$. Both branches reuse one encoder evaluation, so velocity correction adds a shallow decoder rather than a second encoder.

3.3 Bottlenecked Representation Alignment

Representation alignment benefits latent diffusion (Yu et al., 2025), but matching pixel features directly to compressed semantic targets can compromise information needed for reconstruction (Shin et al., 2026). Our **bottlenecked Representation Alignment (bREPA)** applies this supervision through a narrow auxiliary projection while leaving the generative representation full-width.

As illustrated in Fig. 2(a), bREPA takes intermediate encoder features f_t and projects them through a point-wise MLP P_ω with hidden width $d_b \ll D$. The projection preserves spatial positions and matches the teacher feature dimension. For sample i and spatial token n, let $p_{i,n} = P_\omega(f_{t,i,n})$ and let $q_{i,n} = T(x_i)_n$ be the corresponding feature from a frozen DINOv2 teacher. We minimize negative token-wise cosine similarity,

$$\mathcal{L}_{\text{bREPA}} = -\frac{1}{BN}\sum_{i=1}^{B}\sum_{n=1}^{N}\frac{p_{i,n}^\top q_{i,n}}{\|p_{i,n}\|_2\|q_{i,n}\|_2}, \tag{14}$$

where B is the batch size. The bottleneck restricts the alignment projection rather than the width of the generative features. Neither the teacher nor the projector is required at inference.

3.4 Hann Decoder

The Hann Decoder reduces patch-boundary discontinuities by predicting overlapping image regions and blending them with spatial weights.

Cross-scale conditioning. The encoder processes 16×16 image patches, while both decoders initialize their states from 8×8 patches of z_t. A learned $D \to 4d$ projection, where d is the decoder width, maps each encoder token to four conditioning vectors that are rearranged into the corresponding 2×2 fine-grid positions. For 256×256 images, the 16×16 encoder grid thus conditions a 32×32 decoder grid. These features modulate the fine states position-wise together with timestep and class conditioning. Following PixelDiT (Yu et al., 2026), grouped attention compacts each group of four fine states into one attention token and expands its update back to the fine grid. Global attention therefore operates over N tokens while the decoder maintains $4N$ states.

Table 2: **Quantitative comparison on ImageNet** 256×256 **for class-conditional generation.** FID and IS are evaluated with 50K samples.

Method	Pred.	Epochs	#Params	GFLOPs	FID↓	IS↑
Latent-space diffusion						
DiT-XL/2 (Peebles & Xie, 2023)	ϵ	1400	675M	238	2.27	278.2
SiT-XL/2 (Ma et al., 2024)	v	1400	675M	238	2.06	270.3
REPA, SiT-XL/2 (Yu et al., 2025)	v	800	675M	238	1.42	305.7
Pixel-space diffusion						
JiT-B/16 (Li & He, 2026)	x	600	131M	50	3.66	275.1
JiT-L/16 (Li & He, 2026)	x	200	459M	176	2.79	-
k-Diff, JiT-B/16 (Jin & Wang, 2026)	k	600	131M	50	3.64	274.9
PixelREPA-B/16 (Shin et al., 2026)	x	600	131M	50	3.17	284.6
JiT-B/16 + Reg. (Starodubcev et al., 2026)	x	650	131M	53	2.96	–
OiT-B (ours)	$\mathbf{x} + \boldsymbol{\Delta}\mathbf{v}$	**400**	**146M**	**73**	**2.52**	**284.7**
ADM-U (Dhariwal & Nichol, 2021)	ϵ	400	554M	2240	4.59	186.7
DiP-XL/16 (Chen et al., 2026b)	v	600	631M	–	1.79	281.9
PixelGen-XL/16 (Ma et al., 2026b)	x	160	676M	260	1.83	293.6
PixelFlow-XL/4 (Chen et al., 2025c)	v	320	677M	5818	1.98	282.1
DeCo-XL/16 (Ma et al., 2026a)	v	800	682M	245	1.62	301.0
PixNerd-XL/16 (Wang et al., 2026)	v	320	700M	268	1.93	298.0
SiD2, UViT/2 (Hoogeboom et al., 2025)	v	–	–	274	1.72	–
PixelDiT-XL/16 (Yu et al., 2026)	v	320	797M	311	1.61	292.7
JiT-H/16 (Li & He, 2026)	x	600	953M	363	1.86	303.4
PixelREPA-H/16 (Shin et al., 2026)	x	600	953M	363	1.81	317.2
AsymFlow-H/16 (Chen et al., 2026a)	$P\epsilon - x$	600	953M	363	1.57	–
OiT-XL (ours)	$\mathbf{x} + \boldsymbol{\Delta}\mathbf{v}$	**300**	**624M**	**300**	**1.55**	**304.2**

Overlapping reconstruction. Each fine state predicts a 16×16 RGB patch at stride 8, giving neighboring predictions overlapping support. As shown in Fig. 2(c), we weight each patch with a separable periodic Hann window $W(a, b) = w(a)w(b)$, where $w(a) = \frac{1}{2}[1 - \cos(2\pi a/16)]$ for $a \in \{0, \ldots, 15\}$. Let $\mathcal{I}(q)$ index patches covering pixel q, with predictions $P_j(q)$ and weights $W_j(q)$. The output is reconstructed by normalized overlap-add,

$$\hat{Y}(q) = \frac{\sum_{j \in \mathcal{I}(q)} W_j(q) P_j(q)}{\sum_{j \in \mathcal{I}(q)} W_j(q)}. \tag{15}$$

Here $\hat{Y}$ denotes either the clean-image prediction or the velocity residual.

4 Experiments

Experimental setup. We evaluate OiT on class-conditional ImageNet-1K (Deng et al., 2009) generation at 256×256 resolution using FID-50K (Heusel et al., 2017) and Inception Score (Salimans et al., 2016). Unless otherwise specified, evaluation uses EMA checkpoints with 50-step Heun sampling and classifier-free guidance. Detailed configurations are provided in Sec. A.

4.1 Main Results

Table 2 compares OiT with prior class-conditional diffusion models on ImageNet 256×256. OiT-B achieves **2.52 FID** and 284.7 IS after 400 epochs. Its FID is 31.1% lower than JiT-B/16 (3.66) and



Figure 3: **Class-conditional samples on ImageNet** 256×256**.** Samples are generated by OiT-XL.

20.5% lower than PixelREPA-B/16 (3.17), both trained for 600 epochs. Compared with the listed JiT-L/16 result, OiT-B achieves lower FID (2.52 versus 2.79) with fewer parameters (146M versus 459M) and lower forward cost (73.6 versus 176 GFLOPs).

Scaling OiT to the XL size further reduces FID to **1.55**, outperforming PixelDiT-XL/16 (1.61), DeCo-XL/16 (1.62), PixelREPA-H/16 (1.81), and AsymFlow-H/16 (1.57). These results show that ordered velocity correction scales effectively from compact to larger pixel denoisers. Qualitative samples from OiT-XL are shown in Fig. 3. We next examine the contribution of each component.

4.2 Ablation Studies

Component ablation. Table 3 summarizes how each component contributes to OiT. Starting from vanilla x-pred, introducing a nonlinear decoder reduces FID from 9.81 to 8.36, while Hann reconstruction further improves it to 7.79 by alleviating patch-boundary artifacts. Adding bREPA together with a wider decoder brings FID down to 5.84 and substantially improves IS. Finally, nested-prefix velocity correction further reduces FID to 5.68 and raises IS to 211.4, showing that velocity refinement provides complementary gains on top of the improved decoder and representation alignment.

Table 3: **Ablation of components.**

Configuration	FID↓	IS↑
Vanilla x-pred	9.81	147.8
+ Decoder	8.36	162.3
+ Hann Decoder	7.79	165.5
+ bREPA	5.84	206.7
+ Nested-v residual	**5.68**	**211.4**

Table 4: **Ablation of velocity correction.**

(a) Nested masking and correction

Nested	v head	FID↓	IS↑
–	–	5.84	206.7
–	Full	5.81	208.8
✓	–	5.77	209.9
✓	$r = 8$	**5.68**	**211.4**

(b) Correction prefix width

r	FID↓	IS↑
4	5.76	209.2
8	**5.68**	**211.4**
16	5.68	211.1
32	5.79	208.9

(c) Masked fraction

Masked	FID↓	IS↑
0%	5.81	208.8
25%	**5.68**	**211.4**
50%	5.77	209.1

(d) Prefix sampling

Prefix	FID↓	IS↑
Log-spaced	**5.68**	**211.4**
Uniform	5.71	211.3
Fixed-8	6.12	192.3

Ordered velocity correction. Table 4(a) separates nested masking from the residual-head configuration. Without masking, a full-width residual head changes FID from 5.84 to 5.81 and IS from 206.7

to 208.8. Nested masking without a residual head reaches 5.77 FID and 209.9 IS. Their combination with prefix-8 correction gives the best result in this comparison, 5.68 FID and 211.4 IS. Thus, the benefit is not reproduced by simply adding a full-width residual decoder: learning the prefix and using it for correction is more effective in this setting.

Correction prefix width. Table 4(b) varies the number of encoder channels available to the correction branch. Widths 8 and 16 both give 5.68 FID at the reported precision, with $r = 8$ attaining higher IS (211.4 versus 211.1). Reducing the prefix to four channels gives 5.76 FID, while increasing it to 32 gives 5.79. Performance therefore does not improve monotonically with the amount of accessible encoder context. We use $r = 8$, the smaller of the two best-performing prefixes.

Masked fraction. As shown in Table 4(c), masking 25% of training examples yields 5.68 FID, compared with 5.81 for the unmasked full-width residual reference. Increasing the masked fraction to 50% gives 5.77 FID and reduces IS from 211.4 to 209.1. Among the tested settings, the 25% mixture provides the best balance between learning from truncated representations and training the full-width clean branch used at inference.

Prefix sampling. At the same 25% masked fraction, log-spaced sampling over $K \in \{8, 16, 32, 64, 128, 256, 512\}$ performs similarly to uniform sampling, with FIDs of 5.68 and 5.71, respectively (Table 4(d)). In contrast, fixing the masked prefix to $K = 8$ degrades FID to 6.12 and IS to 192.3. The much larger drop under fixed-width training indicates that exposing the clean decoder to multiple prefix widths is more important than the particular sampling schedule.

Table 5: **Ablation of representation alignment.**

(a) bREPA bottleneck width

d_b	FID↓	IS↑
64	3.21	279.0
256	**3.18**	**284.7**
1024	3.22	282.8

(b) Alignment method

Alignment	FID@80ep↓	FID@200ep↓
REPA	**5.69**	3.56
bREPA	5.97	**3.18**
PixelREPA	6.27	3.50

Bottleneck width. Table 5(a) examines the capacity of the auxiliary alignment projector. Increasing d_b from 64 to 256 improves FID from 3.21 to 3.18 and IS from 279.0 to 284.7. A further increase to 1024 gives 3.22 FID and 282.8 IS. The best tested setting is therefore $d_b = 256$, rather than the widest projector. The small variation in FID also indicates that useful alignment does not require a large auxiliary projection. These changes affect the training-time alignment branch, leaving the inference architecture unchanged.

Alignment method. The ranking of alignment methods changes with the training budget (Table 5(b)). At 80 epochs, standard REPA achieves the lowest FID, 5.69, followed by bREPA at 5.97 and PixelREPA at 6.27. After 200 epochs, bREPA reaches 3.18 FID, improving on REPA (3.56) and PixelREPA (3.50) by 0.38 and 0.32, respectively. The early advantage of the wider REPA projector therefore does not persist. This comparison favors evaluating alignment beyond its initial convergence gains and shows that the bottlenecked projection is effective at the longer training budget.

Table 6: **Further analysis of ordered velocity correction.**

(a) PCA vs. learned ordering

Representation	FID↓	IS↑
Fixed PCA-$r8$	3.25	278.4
Nested (ours)	**3.18**	**284.7**

(b) Few-step sampling

Model	Epochs	FID@10↓	IS↑
JiT-B/16	600	7.16	243.2
OiT-B (ours)	400	**5.42**	**289.1**

PCA vs. learned ordering. The PCA analysis in Fig. 1(a) raises a natural question of whether the principal directions can be used directly for velocity correction. With Hann decoder and bREPA, a fixed PCA-$r8$ representation reaches 3.25 FID after 200 epochs, while the learned nested representation achieves 3.18 FID and 284.7 IS (Table 6(a)). The learned representation therefore performs slightly better without relying on dataset-level PCA precomputation.

Few-step sampling. Table 6(b) evaluates the final OiT-B model under a 10-step sampling budget. OiT-B achieves 5.42 FID and 289.1 IS, compared with 7.16 FID and 243.2 IS for JiT-B/16.

5 Related Work

Latent diffusion. Latent diffusion performs denoising in a compact space learned by a pretrained autoencoder, substantially reducing the cost of high-resolution generation (Rombach et al., 2022). This paradigm has become the foundation of Transformer-based diffusion models (Peebles & Xie, 2023; Ma et al., 2024) and large-scale text-to-image systems (Esser et al., 2024; Chen et al., 2024; Ramesh et al., 2021; Wu et al., 2025). Recent work further improves latent generation through representation alignment (Yu et al., 2025; Leng et al., 2025; Singh et al., 2025), stronger compression and structured latent spaces (Chen et al., 2025a;b), and representation autoencoders built from pretrained vision encoders (Zheng et al., 2026). Despite these advances, latent generation remains bounded by the information preserved in the learned representation, motivating approaches that model images directly in pixel space.

Pixel-space diffusion. Early diffusion models operate directly on image pixels (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021), but scaling pixel-space generation remains challenging because the denoiser must process high-dimensional noisy inputs. Recent work improves this setting through architectural and training advances, including improved information routing and scaling strategies (Hoogeboom et al., 2023; 2025), cascaded generation (Chen et al., 2025c), and specialized encoder–decoder designs (Wang et al., 2026; Chen et al., 2026b; Yu et al., 2026). Other approaches exploit frequency-aware architectures (Ma et al., 2026a; Guo et al., 2026), perceptual supervision (Ma et al., 2026b), representation alignment (Shin et al., 2026), or additional register tokens (Starodubcev et al., 2026). OiT follows the encoder–decoder pixel diffusion setting while introducing ordered representations for velocity correction.

Prediction parameterization. Diffusion and flow-based generative models have been trained with noise, score, clean-data, and velocity prediction objectives (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023; Geng et al., 2025). Their behavior can depend strongly on parameterization, preconditioning, and loss weighting (Karras et al., 2022). In pixel space, JiT shows that clean-image prediction can substantially improve both generation and representation learning (Li & He, 2026), while k-Diff adapts the prediction target to the geometry of natural images (Jin & Wang, 2026). AsymFlow (Chen et al., 2026a) instead modifies the target by projecting its noise component into a low-rank subspace. OiT retains full clean-image prediction and learns an additive, full-dimensional velocity correction guided by a compact prefix of an ordered representation.

6 Conclusion

In this work, we studied the trade-off between x-pred and v-pred in pixel diffusion. We found that x-pred learned stronger representations and achieved better long-step generation, yet remained weaker in few-step sampling, with its induced velocity errors concentrated along dominant image directions. Based on this observation, we introduced **OiT**, which learned ordered representations through nested channel masking and used a compact feature prefix to guide a velocity correction while retaining the full representation for clean-image prediction. We further introduced **bREPA** and the **Hann Decoder** as lightweight designs for representation alignment and patch decoding. Across sampling budgets, OiT improved both representation quality and generation, with OiT-B and OiT-XL achieving FID scores of **2.52** and **1.55** on ImageNet 256×256, respectively. These results showed that learned ordered representations provided an effective interface for correcting the velocity induced by x-prediction while preserving its advantages for representation learning and long-step generation.

Reproducibility Statement

Implementation details, training and evaluation settings, and additional analyses are provided in the appendix. Code will be publicly released to facilitate reproduction of our results.

AI Use Statement

Generative AI tools were used for manuscript drafting and editing, literature summarization, figure and table presentation. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final work.

References

Hansheng Chen, Jan Ackermann, Minseo Kim, Gordon Wetzstein, and Leonidas Guibas. Asymmetric flow models. *arXiv preprint arXiv:2605.12964*, 2026a. 2, 6, 9

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024. 9

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *ICLR*, 2025a. 9

Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. In *ICCV*, 2025b. 2, 9

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025c. 6, 9

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. In *CVPR*, 2026b. 6, 9

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 1, 6, 9

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 9

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 9

Zipeng Guo, Lichen Ma, Yu He, Xiaolong Fu, Jingling Fu, Junshi Huang, and Yan Li. Pixelu: A u-shaped transformer for efficient end-to-end pixel diffusion. In *ECCV*, 2026. 9

Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 1978. 2

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. 6

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 9

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 9

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 *CVPR*, 2025. 6, 9

Qing Jin and Chaoyang Wang. Revisiting diffusion model predictions through dimensionality. *arXiv preprint arXiv:2601.21419*, 2026. 6, 9

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 9

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. In *ICCV*, 2025. 9

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, 2026. 1, 3, 6, 9, 14

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 1, 3, 9

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 1, 3, 9

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 *ECCV*, 2024. 1, 6, 9

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *CVPR*, 2026a. 6, 9

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026b. 6, 9, 14

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 9

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 6, 9

Nicolas Pielawski and Carolina Wählby. Introducing hann windows for reducing edge-effects in patch-based image segmentation. *PloS one*, 2020. 2

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 9

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *ICML*, 2014. 2, 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 9

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 6

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. In *ECCV*, 2026. 1, 2, 5, 6, 9

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *ICLR*, 2025. 9

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. 9

Nikita Starodubcev, Ilia Sudakov, Ilya Drobyshevskiy, Artem Babenko, and Dmitry Baranchuk. Registers matter for pixel-space diffusion transformers. *arXiv preprint arXiv:2605.16147*, 2026. 6, 9

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *ICLR*, 2026. 1, 6, 9

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1, 9

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 *ICLR*, 2025. 2, 5, 6, 9

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *CVPR*, 2026. 1, 5, 6, 9

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026. 1, 9

A Implementation Details

Architecture. The architectural configurations of OiT-B and OiT-XL are summarized in Tab. 7.

Table 7: **Architecture configurations of OiT.**

Module	Configuration	OiT-B	OiT-XL
Encoder	Patch size	16	16
	Depth	12	22
	Hidden size	768	1024
x decoder	Token stride	8	8
	Depth	2	4
	Hidden size	384	768
Δv decoder	Token stride	8	8
	Depth	2	2
	Hidden size	384	768
Hann output	Output patch size	16	16
	Output stride	8	8
#Params		146M	624M
GFLOPs		73.6	300

Ordered representation and velocity correction. For OiT-B, the encoder representation has width $D = 768$. During training, the full representation is retained with probability 0.75. Otherwise, we sample

$$K \in \{8, 16, 32, 64, 128, 256, 512\},$$

retain the first K channels, and replace the remaining channels with a learned null embedding. The same channel mask is applied to all spatial tokens. The clean decoder is trained directly from the sampled representation, while full-width features are used at inference.

The correction branch always uses the first $r = 8$ channels. The clean prediction from the same forward pass is used to construct the residual target. We set the endpoint floor to $t_\epsilon = 0.05$ and use the correction gate $g(t) = t^2$. The residual target is stop-gradient. The noisy-image state and global conditioning passed to the correction branch are also detached, while the semantic prefix remains differentiable. Thus, residual supervision updates the correction decoder and propagates through the leading encoder channels without modifying the clean decoder through the residual branch.

Representation alignment. bREPA is applied to intermediate encoder features, taken after the fourth block for OiT-B and the eighth block for OiT-XL. We use DINOv2-B/14 patch features as semantic targets, obtained from the clean image resized to 224×224. Both the encoder and teacher produce a 16×16 token grid, allowing direct spatial correspondence. The auxiliary projector is a point-wise bottleneck MLP of the form

$$D \to d_b \to d_b \to D_T,$$

where D is the encoder width and $D_T = 768$ is the DINOv2-B/14 feature dimension. SiLU activations are used between linear layers. We use $d_b = 256$ for the final configuration, as selected in Tab. 5. Projected features and teacher features are ℓ_2-normalized and optimized with token-wise cosine similarity. The alignment branch is used only during training and does not alter the inference architecture.

Hann Decoder. Both clean-image prediction and velocity correction use overlapping patch reconstruction. Each decoder state predicts a 16×16 RGB patch on a stride-8 grid, resulting in 50% overlap between neighboring predictions. We use the periodic Hann window

$$w(i) = \frac{1}{2}\left(1 - \cos\frac{2\pi i}{16}\right), \qquad i = 0, \ldots, 15,$$

and construct the two-dimensional weight as $W(i, j) = w(i)w(j)$. Overlapping predictions are accumulated with the same weights and normalized pixel-wise by the accumulated window values. In implementation, reconstruction is performed with a 16×16 folding kernel, stride 8, and padding 4, with the normalization denominator clamped to 10^{-6} for numerical stability.

Optimization. OiT-B follows the optimization recipe of JiT (Li & He, 2026), using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, a learning rate of 2×10^{-4}, zero weight decay, and a five-epoch linear warmup followed by a constant learning rate. OiT-XL instead follows the optimization recipe of PixelGen (Ma et al., 2026b). In addition to the objectives used for OiT-B, OiT-XL incorporates the LPIPS and P-DINO losses following PixelGen. Both models are trained with exponential moving average with decay 0.9999. The timestep sampler uses $P_{\text{mean}} = -0.8$ and $P_{\text{std}} = 0.8$, with unit noise scale.

Evaluation. Unless otherwise specified, we evaluate EMA checkpoints using 50-step Heun sampling and report FID-50K and Inception Score on ImageNet 256×256. Classifier-free guidance is set to 2.75 for OiT-B and 2.25 for OiT-XL. For the few-step evaluation in Tab. 6, we use 10 sampling steps while keeping the remaining evaluation protocol unchanged.