SEMPIX: LEARNING SEMANTIC FLOWS FOR PIXEL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel diffusion generates images directly in pixel space without a separately trained autoencoder. Yet a single pixel trajectory must model object structure and spatial layout together with the diverse textures, colors, and illumination that realize them. Pretrained visual representations reduce sensitivity to such appearance variation and organize images by content and spatial structure, providing a more direct target for the visual structure required by pixel generation. We introduce *SemPix*, a coupled semantic–pixel flow framework that models pretrained visual representations as an explicit generative state for native pixel synthesis, rather than as auxiliary supervision alone. A semantic flow learns the distribution of dense pretrained features, while a pixel flow generates the complete image from independent noise under spatially aligned modulation by the generated state. This coupling makes semantic structure part of the generative process while preserving stochastic generation in native pixel space. The two flows are trained end-to-end through predicted semantic states, combining representation supervision with pixel-generation gradients that adapt the semantic state for image synthesis. On class-conditional ImageNet 256×256, SemPix converges substantially faster than native pixel diffusion baselines without guidance and achieves competitive generation quality against state-of-the-art pixel generators after full training. These results show that modeling semantic structure as an explicit generative state can reduce the optimization burden of native pixel diffusion.

1 INTRODUCTION

Diffusion models generate images along a learned trajectory from noise to the data distribution (Ho et al., 2020; Lipman et al., 2023), with the choice of modeling space determining both computational cost and the visual information that the denoiser must recover. Latent diffusion operates in an autoencoder-defined space to reduce computation (Rombach et al., 2022; Peebles & Xie, 2023; Esser et al., 2024), whereas pixel diffusion treats the image itself as the clean target and avoids a separately trained reconstruction decoder (Chen et al., 2025; Wang et al., 2025a; Li & He, 2025). Recent advances have made direct pixel modeling increasingly effective (Ma et al., 2025; Li & He, 2025; Yu et al., 2025b; Ma et al., 2026). Yet a pixel trajectory must still model object identity and spatial layout together with the colors, textures, and illumination that realize them. Their joint expression in pixel values couples the recovery of semantic organization with the modeling of detailed appearance.

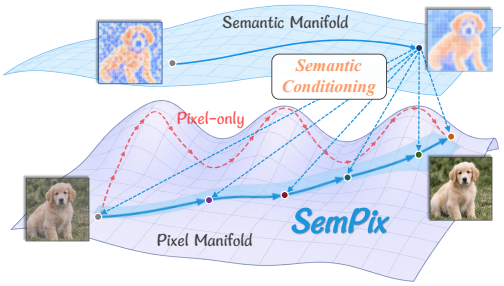

Figure 1: **Conceptual view of SemPix.** Pixel-only generation resolves semantic structure and appearance along a single pixel trajectory. SemPix instead generates a state in a pretrained representation space to condition pixel diffusion.

Pretrained visual representations offer a complementary learning target. Their features preserve object content and spatial organization while reducing sensitivity to photometric and geometric variation (Zhang et al., 2018; Mahendran & Vedaldi, 2015; Siméoni et al., 2025), making visual structure

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.

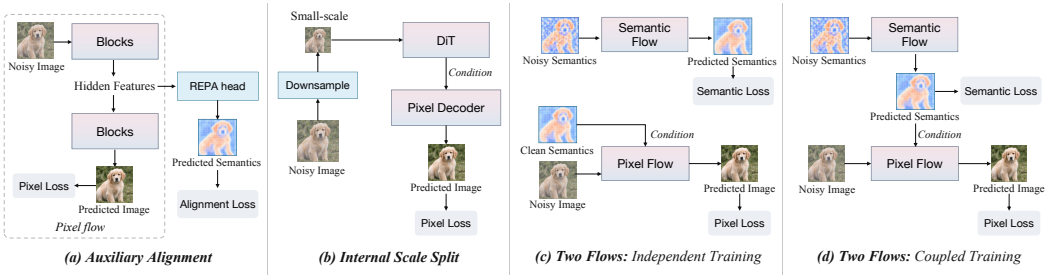

Figure 2: **Paradigms for pixel generation.** (a) Auxiliary alignment supervises hidden pixel features. (b) Internal scale splitting organizes computation across image scales. (c) *Independent training* uses encoder-derived pixel conditions with optional noise; its pixel loss updates only the pixel flow. (d) *Coupled training* uses a one-pass clean-state prediction, allowing image-loss gradients to update both flows while retaining semantic supervision.

more directly accessible than through pixel denoising alone. Accordingly, aligning diffusion features with pretrained representations accelerates training (Yu et al., 2025a; Shin et al., 2026). Their utility is not limited to feature supervision: diffusion over pretrained representations supports high-quality synthesis and converges faster than VAE-based latent baselines (Zheng et al., 2025). These results indicate that pretrained features provide both an informative target for semantic learning and a distribution that can itself be modeled generatively.

These observations motivate the semantic–pixel factorization illustrated in Figure 1. Rather than requiring a single pixel trajectory to resolve semantic structure together with detailed appearance, the factorization models semantic structure as an explicit random variable in native pixel generation. A semantic flow models the distribution of dense pretrained representations, while a conditional pixel flow models the distribution of images compatible with a sampled state. Encoder-derived targets train the semantic flow to predict representations that encode object content and spatial organization. The generated state makes this structure available to the pixel flow even when the noised image contains little reliable visual information. Semantic structure thereby becomes part of the sampled image model, rather than only a constraint on hidden activations (Yu et al., 2025a; Shin et al., 2026) or a condition constructed within a pixel-valued hierarchy (Ma et al., 2025; Yu et al., 2025b) (Figure 2).

We instantiate this factorization as *SemPix*, which couples semantic and pixel flows. The semantic flow generates a dense spatial feature map that serves as a *generative semantic state*. Starting from independent noise, the pixel flow uses a spatially aligned form of adaptive normalization (Peebles & Xie, 2023) to condition every transformer block on this state while generating the complete image. We couple the two flows end-to-end by conditioning the pixel flow on predicted semantic states during joint training. This aligns the source of semantic conditions between training and inference, teaching the pixel flow to accommodate errors in the semantic generator's predictions. Image-generation gradients also propagate through the condition into the semantic flow, adapting its predictions for pixel synthesis while representation supervision anchors them to the pretrained feature distribution. An encoder-feature conditioning stage initializes the representation-to-pixel mapping before this coupled training. At inference, the semantic flow generates the spatial state from noise, and the pixel flow samples its pixel realization without evaluating the visual encoder.

We evaluate SemPix on class-conditional ImageNet generation (Deng et al., 2009) at 256×256 resolution. SemPix converges substantially faster than native pixel diffusion baselines without guidance, suggesting that the generated semantic state provides useful content and spatial organization for pixel learning. End-to-end coupling further adapts this state to image synthesis by allowing pixel-generation gradients to update the semantic flow, improving generation quality. With full training, SemPix achieves competitive generation quality against state-of-the-art pixel generators (Li & He, 2025; Ma et al., 2025; Yu et al., 2025b; Ma et al., 2026). Our main contributions are as follows.

(1) We introduce SemPix to address the burden of learning semantic structure and detailed appearance within a single pixel trajectory. Its semantic–pixel factorization models visual structure in a pretrained representation space and uses the generated state to condition native pixel synthesis.

(2) We couple semantic and pixel flows end-to-end through predicted semantic states, aligning the source of pixel conditioning between training and inference. Image-generation gradients further adapt the semantic flow for pixel synthesis, while a frozen encoder provides representation targets.

(3) On class-conditional ImageNet 256×256, SemPix converges substantially faster than native pixel diffusion baselines without guidance and achieves competitive generation quality with state-of-the-art pixel generators after full training.

2 Related Work

Pixel diffusion and hierarchical computation. Diffusion and score-based models generate images directly from noise (Ho et al., 2020; Song et al., 2021), with improvements to noise modeling, denoiser architectures, and sampling increasing efficiency and fidelity (Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Karras et al., 2022). While latent diffusion reduces computation through an autoencoder (Rombach et al., 2022) and supports scalable transformer and flow-based models (Peebles & Xie, 2023; Lipman et al., 2023; Ma et al., 2024), native pixel methods retain the image as the generative endpoint. Simple diffusion and SiD2 improve high-resolution training through noise schedules, architectures, and objectives (Hoogeboom et al., 2023; 2025); HDiT uses multi-scale computation (Crowson et al., 2024), and cascaded diffusion and PixelFlow organize generation across resolutions (Ho et al., 2022; Chen et al., 2025). PixNerd predicts pixel velocities through patch-wise neural fields (Wang et al., 2025a), while JiT directly predicts clean images (Li & He, 2025). DeCo, PixelDiT, and PixelU further allocate computation across frequencies or spatial scales (Ma et al., 2025; Yu et al., 2025b; Guo et al., 2026). Complementing these architectural choices, representation supervision improves the learning signal: REPA aligns hidden features with frozen encoder targets (Yu et al., 2025a), HASTE refines alignment through attention matching and early stopping (Wang et al., 2025c), and iREPA emphasizes spatial structure (Singh et al., 2025). In pixel space, PixelREPA addresses failures of direct alignment in JiT with a masked adapter (Shin et al., 2026), and PixelGen introduces perceptual supervision (Ma et al., 2026). SemPix builds on these efforts to ease pixel learning, but explicitly samples a semantic state from its own prior and uses it throughout pixel generation, beyond organizing computation or supervising features within a pixel denoiser.

Representation-conditioned image generation. Pretrained visual features provide content and spatial information, as shown by feature inversion, perceptual similarity, and the DINO family (Mahendran & Vedaldi, 2015; Zhang et al., 2018; Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025). Using such representations to condition image synthesis has substantial precedent. Diffusion Autoencoders learn semantic codes with a conditional diffusion decoder (Preechakul et al., 2022), while RCDM visualizes self-supervised features through conditional sampling (Bordes et al., 2022). Self-conditioned diffusion uses feature-derived cluster labels (Bao et al., 2022); unCLIP and RCG instead learn representation priors whose samples condition image generation (Ramesh et al., 2022; Li et al., 2024). RAE models pretrained dense features with diffusion and reconstructs images through a deterministic decoder (Zheng et al., 2025), while RAEv2 shows that representation-space generation and auxiliary alignment can be complementary (Singh et al., 2026). PiD provides a closely related pixel diffusion decoder for pretrained latent generators (Lu et al., 2026). The conditioning mechanism also has established foundations in feature-wise modulation (Perez et al., 2018), spatially adaptive normalization (Park et al., 2019), and adaptive normalization in DiT and PixelDiT (Peebles & Xie, 2023; Yu et al., 2025b). SemPix combines a generated dense representation with spatial modulation of a native-resolution pixel flow, which synthesizes the complete image from independent noise. Conditioning augmentation in cascaded diffusion (Ho et al., 2022) motivates our noise-augmented encoder-condition control; our main focus is learning the semantic prior through its effect on conditional pixel synthesis.

Joint semantic–image modeling and end-to-end training. Joint generative modeling gives semantic representations a role during sampling as well as training. ReDi and REG jointly denoise image latents and pretrained features (Kouzelis et al., 2025; Wu et al., 2025); Semantic-First Diffusion orders semantic and texture denoising asynchronously (Pan et al., 2025), and REGLUE combines global and local semantics for joint latent diffusion (Petsangourakis et al., 2026). In pixel space, Latent Forcing jointly processes representation and pixel tokens with modality-specific schedules;

non-overlapping schedules also admit a semantic-prior/conditional-image factorization (Baade et al., 2026). CoReDi adapts its representation space through a learned projection jointly with image–feature diffusion, including a pixel-space variant (Kouzelis et al., 2026). REPA-E further demonstrates end-to-end adaptation by jointly tuning a VAE and latent diffusion model (Leng et al., 2025). Within this line of work, SemPix studies a specific interface between separately parameterized semantic and pixel flows: a one-pass clean prediction from a paired noisy semantic target conditions the pixel flow during training, and image losses propagate through that prediction. The encoder-defined target space remains fixed, while feedback adapts the semantic generator within that space. At inference, the semantic flow finishes before pixel generation begins. The detached-condition ablation in Section 4.4 isolates this gradient connection from the use of a predicted condition itself.

3 METHOD

3.1 TWO-FLOW FORMULATION

SemPix separates semantic-state generation from conditional image synthesis, then couples their learning through the semantic prediction used for pixel generation (Figure 3). For an image $x \in \mathbb{R}^{3\times H\times W}$ with label y, a frozen visual encoder E supplies the semantic target $z = E(x) \in \mathbb{R}^{C_z\times H_z\times W_z}$ (Siméoni et al., 2025). Here E includes fixed preprocessing and feature normalization. The two flows define

$$p_{\theta,\phi}(x \mid y) = \int p_\phi(x \mid z, y)\, p_\theta(z \mid y)\, \mathrm{d}z. \tag{1}$$

The Semantic flow F_θ induces $p_\theta(z \mid y)$ by generating dense semantic states. The Pixel flow G_ϕ induces $p_\phi(x \mid z, y)$ through a conditional stochastic process at native image resolution, retaining variation left unspecified by the semantic state. We denote its semantic condition by $s \in \mathbb{R}^{C_z\times H_z\times W_z}$.

The Semantic flow uses a Decoupled Diffusion Transformer (DDT) (Wang et al., 2025b), following RAE (Zheng et al., 2025). Its shallow diffusion head predicts a clean semantic map from noisy tokens and deep backbone features. The Pixel flow uses this prediction to spatially modulate its tokens through AdaLN in each block, building on spatial and adaptive normalization in image generators (Park et al., 2019; Peebles & Xie, 2023; Yu et al., 2025b).

Both flows use conditional flow matching (Lipman et al., 2023) with clean-state prediction (Li & He, 2025). For a state $\xi \in \{z, x\}$ and Gaussian noise $\epsilon_\xi \sim \mathcal{N}(0, I)$, we construct the noisy input and convert a clean prediction to a velocity as

$$\xi_t = t\xi + (1-t)\epsilon_\xi, \qquad \mathcal{V}(\widehat{\xi}, \xi_t, t) = \frac{\widehat{\xi} - \xi_t}{\max(1-t, \delta)}, \tag{2}$$

where $t \in [0, 1]$ runs from noise to data and $\delta > 0$ stabilizes the denominator. The target velocity uses the same conversion and equals $\xi - \epsilon_\xi$ when $1 - t \geq \delta$. The velocity loss is therefore a weighted clean-state MSE with $\omega(t) = \max(1-t, \delta)^{-2}$. We use $\|a\|^2_{\mathrm{av}} = \|a\|_2^2 / \dim(a)$ for coordinate-averaged squared error. Semantic and pixel inputs use independent noise draws and timesteps t_z, t_x; expectations below include training pairs and all random draws.

3.2 COUPLED TRAINING THROUGH PREDICTED SEMANTICS

Matching encoder features alone gives the Semantic flow no direct feedback on how its predictions affect image synthesis. Coupled training (CT) introduces this feedback by using the clean semantic prediction as a differentiable condition for the Pixel flow:

$$s_{\mathrm{pred}} = \widehat{z} = F_\theta(z_{t_z}, t_z, y), \qquad \widehat{x} = G_\phi(x_{t_x}, t_x, y, s_{\mathrm{pred}}). \tag{3}$$

Both noisy inputs come from the same training image, so the semantic prediction remains paired with the pixel target. Computing the condition requires one semantic forward pass, without integrating a sampling trajectory. Clean-state prediction gives the condition the same meaning at every noise level: an estimate of the encoder-defined semantic target.

Connecting the learning objectives. We keep s_{pred} attached to the computation graph. Image losses can then update the Semantic flow through the Pixel flow and its spatial conditioning path,

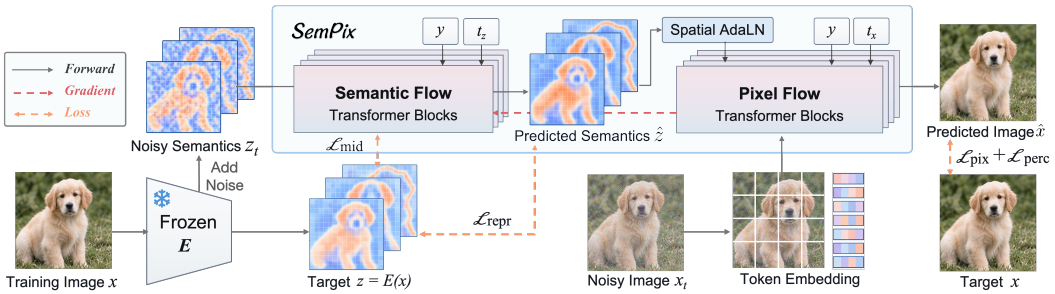

Figure 3: **Coupled training of SemPix.** A single semantic prediction $\hat{z}$ is supervised by the frozen encoder target $z = E(x)$ and conditions the Pixel flow through spatial AdaLN. The pixel and perceptual losses, $\mathcal{L}_{\text{pix}}$ and $\mathcal{L}_{\text{perc}}$, send gradients through this condition to the Semantic flow (red dashed arrow). Intermediate supervision $\mathcal{L}_{\text{mid}}$ targets an earlier semantic prediction. The diagram uses $\mathcal{L}_{\text{repr}}$ for $\mathcal{L}_{\text{sem}}$.

as indicated by the red dashed arrow in Figure 3. This gradient connection defines CT. Supplying a predicted condition alone does not establish it: detaching that condition leaves the forward predictions unchanged but prevents image losses from updating F_θ. The frozen encoder E provides the target and is never updated by either loss.

A shared prediction, two supervision signals. The semantic objective measures agreement with the encoder target:

$$\mathcal{L}_{\text{sem}} = \mathbb{E}\big[\omega(t_z)\|\hat{z} - z\|_{\text{av}}^2\big]\,. \tag{4}$$

We also supervise an intermediate prediction of the same clean semantic target. An auxiliary head H_ψ maps the hidden state $h^{(\ell)}$ at layer ℓ of the semantic backbone to

$$\hat{z}^{(\ell)} = H_\psi\Big(h^{(\ell)}, t_z\Big)\,, \qquad \mathcal{L}_{\text{mid}} = \mathbb{E}\left\|\hat{z}^{(\ell)} - z\right\|_{\text{av}}^2\,. \tag{5}$$

This intermediate output receives its own supervision; the main prediction $\hat{z}$ remains the condition passed to the Pixel flow.

Image supervision evaluates the main semantic prediction through the conditional image generator. It combines the pixel objective

$$\mathcal{L}_{\text{pix}} = \mathbb{E}\big[\omega(t_x)\|\hat{x} - x\|_{\text{av}}^2\big] \tag{6}$$

with $\mathcal{L}_{\text{perc}} = \mathbb{E}[\text{LPIPS}(x, \hat{x})]$, computed using a frozen VGG-based network (Zhang et al., 2018). The full objective is

$$\mathcal{L} = \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{mid}}\mathcal{L}_{\text{mid}} + \lambda_{\text{pix}}\mathcal{L}_{\text{pix}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}}. \tag{7}$$

The semantic terms retain the pretrained representation as a fixed reference; the image terms train the semantic prediction according to its effect on pixel synthesis. CT thus learns the Semantic flow under both criteria, while keeping the encoder-defined feature space fixed.

3.3 Two-Stage Sampling

Generation runs the Semantic flow to completion before starting the Pixel flow. We draw $z_0 \sim \mathcal{N}(0, I)$ and integrate the Semantic flow for N_z steps to obtain $\tilde{z} = z_{N_z}$. We then draw independent pixel noise $x_0 \sim \mathcal{N}(0, I)$ and hold $\tilde{z}$ fixed as the condition for all N_x Pixel flow steps. The final pixel state x_{N_x} is the generated image. The frozen encoder is not used during generation.

Each step predicts a clean state, converts it to a velocity using Equation 2, and applies an Euler update. Writing ξ_k for the current state in either flow and $\widehat{\xi}_k$ for its clean prediction, the update on that flow's time grid is

$$\xi_{k+1} = \xi_k + (t_{k+1} - t_k)\,\mathcal{V}(\widehat{\xi}_k, \xi_k, t_k), \qquad \xi \in \{z, x\}, \tag{8}$$

with $0 = t_0 < \cdots < t_N = 1$ and $N = N_z$ or N_x. During semantic sampling, $\widehat{z}_k = F_\theta(z_k, t_k, y)$; during pixel sampling, $\widehat{x}_k = G_\phi(x_k, t_k, y, \tilde{z})$. Appendix A.4 specifies the Pixel flow time grid.

Guided sampling modifies clean predictions before velocity conversion, using internal Semantic flow guidance (Singh et al., 2026) and Pixel flow classifier-free guidance (Ho & Salimans, 2022):

$$\widehat{z}_{\text{guided}} = \widehat{z}^{(\ell)} + w_z\big(\widehat{z} - \widehat{z}^{(\ell)}\big), \tag{9}$$

$$\widehat{x}_{\text{guided}} = \widehat{x}_{\text{ref}} + w_x\big(\widehat{x}_{\text{cond}} - \widehat{x}_{\text{ref}}\big). \tag{10}$$

The semantic predictions share one forward pass. For the Pixel flow, $\widehat{x}_{\text{cond}}$ uses both y and $\widetilde{z}$, while $\widehat{x}_{\text{ref}}$ drops either or both conditions. Table 4 lists the guidance settings. With 40 Semantic flow and 10 Pixel flow Euler updates, both SemPix models have a nominal guided budget of $40+10\times 2 = 60$ network evaluations.

4 Experiments

We evaluate training progress and generation quality, then isolate the effects of flow coupling, initialization, condition augmentation, capacity allocation, and sampling-time allocation.

4.1 Evaluation Protocol

Dataset and metrics. We study class-conditional generation on ImageNet-1K (Deng et al., 2009) at 256×256 resolution. We report Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016), reporting unguided and guided results separately where available. All SemPix FID evaluations use 50,000 generated images. Published baselines retain their source evaluation protocols (Appendix A.2).

Model and optimization. The base SemPix-L/16 model has 468M parameters; its guided configuration has 471M, including the REPA head retained for semantic prediction. SemPix-H/16 has 970M parameters. A frozen DINOv3 ViT-L/16 encoder (Siméoni et al., 2025) supplies $1024 \times 16 \times 16$ semantic targets. In SemPix-L/16, the Semantic flow uses a DDT with 16 encoder blocks and two diffusion-decoder blocks; the Pixel flow uses six transformer blocks with 16×16 pixel patches. Both flows have width 1024 and 16 attention heads. We use AdamW with learning rate 10^{-4} and EMA decay 0.9999; further optimization details appear in Appendix A.1.

Training and sampling budgets. Independent training (IT) keeps image-loss gradients within the Pixel flow; coupled training (CT) propagates them through predicted conditions into the Semantic flow. We report early optimization at approximately 40 epochs and main results at 80 epochs for L/16 and 160 epochs for H/16. The controlled training and capacity ablations use a separate 40-epoch protocol, detailed in Section 4.4. All SemPix evaluations use 40 Semantic flow steps followed by 10 Pixel flow steps with Euler integration. The early comparison and all ablations use no guidance; guided configurations appear in Table 4. The Pixel flow time shift is 10 except in the time-shift sweep. Training epochs measure image exposure; sampling steps and network evaluations measure different aspects of the generation budget.

4.2 Optimization Behavior

SemPix reaches lower FID early in training and continues to improve with additional epochs. Figure 4 compares unguided L-size models from SemPix, JiT, JiT + REPA, and DeCo. SemPix's FID decreases from 8.19 at 20 epochs to 3.87 at 40 epochs and 3.27 at 80 epochs. Its 20-epoch FID is already below those of JiT and JiT + REPA at 80 epochs (25.18 and 18.60, respectively). This comparison establishes earlier progress in FID as a function of training exposure; the models have different computational costs.

Table 1 extends the comparison to a broader set of pixel generators at approximately 40 epochs, including CoReDi and JiT + iREPA. SemPix achieves FID 3.87 and IS 195.0, compared with 7.53 and 131.7 for PixelGen. This is a 48.6% reduction in FID relative to the strongest baseline in the table. These comparisons retain each method's architecture, supervision, and sampler; the ablations below isolate design choices within SemPix.

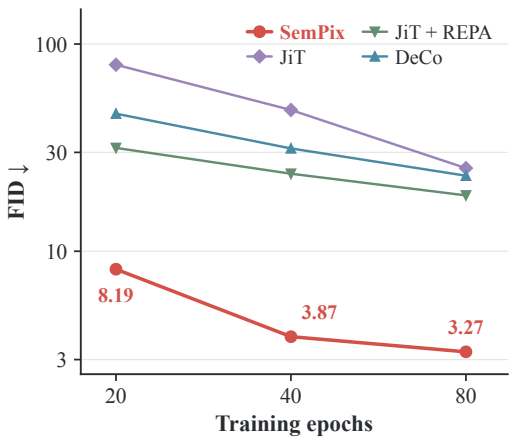

Figure 4: Training progress without guidance. Both axes are logarithmic.

Table 1: Unguided L-size pixel generators at ∼40 training epochs. *Steps* counts sampling updates; 40+10 denotes Semantic + Pixel flow steps. A dash indicates unreported IS.

Method	Params	Steps	FID↓	IS↑
PixelDDT-L/16	434M	50	46.37	36.2
PixNerd-L/16	459M	50	37.49	43.0
PixelFlow-L/4	459M	100	54.33	24.7
DeCo-L/16	426M	50	31.35	48.4
CoReDi-L/16	426M	50	21.5	-
JiT-L/16	459M	50	48.06	30.0
JiT-L/16 + REPA	459M	50	23.67	64.1
JiT-L/16 + iREPA	459M	50	29.19	50.7
PixelGen-L/16	459M	50	7.53	131.7
SemPix-L/16	468M	40 + 10	**3.87**	**195.2**

Table 2: Unguided and guided class-conditional generation on ImageNet 256×256. Params, Sampler, and NFE describe guided inference; NFE includes CFG branches. **SemPix uses Euler with $40 + 10 \times 2 = 60$ nominal NFE**. †: nominal full-CFG upper bound; ≈: converted epochs; dash: unavailable. Sources and protocol details are in Appendix A.2.

	Method	Params	Epochs	Sampler	NFE↓	Unguided		Guided	
						FID↓	IS↑	FID↓	IS↑
Latent	U-ViT-L/2 (Bao et al., 2023)	287M + 50M	≈240	DP/50	100	-	-	3.40	219.9
	MaskDiT (Zheng et al., 2024)	730M + 50M	-	H/40	160	5.69	178.0	2.28	276.6
	DiT-XL/2 (Peebles & Xie, 2023)	675M + 50M	≈1400	D/250	500	9.62	121.5	2.27	278.2
	SiT-XL/2 (Ma et al., 2024)	675M + 50M	≈1400	EM/250	500	8.26	131.7	2.06	277.5
	MDT-XL/2 (Gao et al., 2023)	676M + 50M	≈1300	D/250	500	6.23	143.0	1.79	283.0
	SiT-XL/2 + REPA (Yu et al., 2025a)	675M + 50M	200	EM/250	500	6.40	-	1.96	264.0
	SiT-XL/2 + REPA (Yu et al., 2025a)	675M + 50M	800	EM/250	500†	5.90	157.8	1.42	305.7
	DDT-XL/2 (Wang et al., 2025b)	675M + 50M	80	E/250	500†	6.62	135.2	1.52	263.7
Pixel	PixNerd-L/16 (Wang et al., 2025a)	458M	160	E/50	100†	-	-	2.64	297.0
	JiT-L/16 (Li & He, 2025)	459M	600	H/50	200†	13.85	104.2	2.36	298.5
	JiT-H/16 (Li & He, 2025)	953M	600	H/50	200†	7.15	151.7	1.86	303.4
	PixelREPA-L/16 (Shin et al., 2026)	459M	600	H/50	200†	-	-	2.11	309.5
	PixelREPA-H/16 (Shin et al., 2026)	953M	600	H/50	200†	-	-	1.81	317.2
	PixelU-H/16 (Guo et al., 2026)	1168M	600	H/50	200†	-	-	1.63	305.9
	LF-DIT-L (Baade et al., 2026)	465M	200	-	-	7.20	-	2.48	-
	SemPix-L/16	471M	80	E/(40 + 10)	**60**†	3.27	202.9	2.17	245.0
	SemPix-H/16	970M	160	E/(40 + 10)	**60**†	2.40	225.2	1.83	250.4

4.3 Main Generation Results

Table 2 compares SemPix with latent- and pixel-space generators under their reported training and evaluation protocols. Parameter counts include the generator and, where required, a separate pre-trained image decoder, shown as two terms rounded to the nearest million. SemPix's count includes both flows and any retained prediction head; no additional image decoder is required. Appendix A.2 documents sources, parameter accounting, and sampling conventions.

Unguided generation. SemPix-L/16 achieves FID 3.27 after 80 epochs, and SemPix-H/16 reaches 2.40 after 160 epochs. Both improve on the corresponding JiT results at 600 epochs (13.85 and 7.15), supporting effective generation with less training exposure.



Figure 5: Qualitative results of class-to-image generation with SemPix-H/16. All images are generated at 256×256 resolution.

Table 3: Training and conditioning ablations.

(a) Training strategy.

Strategy	FID↓	IS↑
Encoded IT	4.56	185.06
CT	**4.29**	**186.43**

(b) Gradient feedback.

Feedback	FID↓	IS↑
Detached	4.39	185.54
Enabled	**4.29**	**186.43**

(c) IT warm-up.

IT (ep)	FID↓	IS↑
0	4.46	184.74
20	**4.29**	**186.43**

(d) Condition noise.

Noise	FID↓	IS↑
None	4.75	176.99
Added	**4.56**	**185.06**

Guided generation. SemPix-L/16 achieves FID 2.17 after 80 epochs, compared with 2.36 for JiT-L/16 after 600 epochs. SemPix-H/16 reaches FID 1.83 after 160 epochs, close to JiT-H/16 (1.86) and PixelREPA-H/16 (1.81) after 600 epochs. These results show competitive FID at shorter training budgets, although guided IS remains lower than for the corresponding JiT and PixelREPA models. Both SemPix sizes use 60 nominal guided network evaluations, comprising 40 Semantic flow evaluations and 10×2 Pixel flow evaluations. This is 70% below the nominal 200-NFE budget of 50-step Heun with CFG; actual runtime also depends on each model's cost per evaluation.

Qualitative results. Figure 5 illustrates recognizable object shapes and fine appearance details across classes, including petal patterns, reptile scales, and animal fur.

4.4 Ablation Studies

Table 3 examines flow coupling, image-gradient feedback, initialization, and condition augmentation. Figures 6 and 7 examine capacity and sampling-time allocation. The training and capacity ablations use 468M L-size models for 40 epochs, without REPA or $\mathcal{L}_{\text{mid}}$. The semantic, pixel, and perceptual losses have unit weights. Unless varied, CT uses 20 epochs of IT initialization followed by 20 epochs of coupled training. All ablations use unguided 40+10-step Euler sampling and 50,000 generated images, with Pixel flow shift 10 unless varied. Evaluation always uses sampled semantic states; encoder-derived conditions are restricted to training. The time-shift sweep uses the 40-epoch checkpoint from Table 1, trained with $\mathcal{L}_{\text{mid}}$. We therefore compare settings within this sweep separately from the training and capacity ablations.

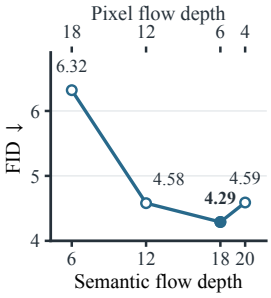

Figure 6: FID vs. flow depth.

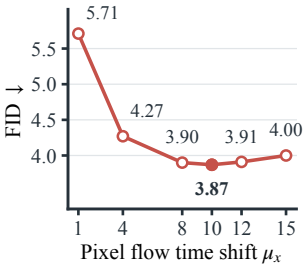

Figure 7: Pixel time shift.

Independent vs. coupled paradigms. Table 3(a) compares the independent and coupled paradigms introduced in Section 1. IT uses noise-augmented encoder conditions throughout train-

ing; CT switches to predicted conditions after initialization and allows image losses to update both flows. CT improves FID from 4.56 to 4.29 and IS from 185.06 to 186.43. This comparison changes both the condition source and the gradient connection. Panel (b) isolates the latter by retaining CT's predicted conditions and initialization while detaching the condition from the computation graph. Detachment blocks both pixel-loss and LPIPS gradients to the Semantic flow, while preserving semantic supervision. Restoring this connection improves FID from 4.39 to 4.29 and IS from 185.54 to 186.43. The observed improvement supports training semantic predictions through their downstream image-generation objective.

Initialization and condition augmentation. IT initialization first trains the Pixel flow on encoder-derived conditions while the Semantic flow learns its own objective. CT then replaces these conditions with attached predictions s_{pred}. At a fixed total budget of 40 epochs, 20 epochs of IT followed by 20 epochs of CT outperform CT from scratch (Table 3(c)). FID improves from 4.46 to 4.29, and IS from 184.74 to 186.43.

During IT, we augment the encoder-derived condition z as $s_{\text{enc}}(\rho) = (1-\rho)z + \rho\epsilon_s$, with $\rho \sim \mathcal{U}(0, \rho_{\max})$ and $\epsilon_s \sim \mathcal{N}(0, I)$. The noise has the shape of z and is independent of both flows' input noises. Setting $\rho = 0$ recovers the clean condition. Table 3(d) isolates this augmentation in 40-epoch IT runs, where image losses update only the Pixel flow. Using $\rho_{\max} = 0.8$ improves FID from 4.75 to 4.56 and IS from 176.99 to 185.06. This result supports noise augmentation during IT when inference uses generated semantic conditions.

Capacity allocation. Figure 6 compares four allocations of a fixed total of 24 transformer blocks. The notation $(a+2)/b$ denotes a Semantic flow encoder blocks, two diffusion-decoder blocks, and b Pixel flow blocks. The bottom axis shows total Semantic flow depth $a+2$; the top axis shows Pixel flow depth b. Increasing a from 4 to 10 to 16 improves FID from 6.32 to 4.58 to 4.29. Increasing it further to 18, leaving four Pixel flow blocks, raises FID to 4.59. The $(16+2)/6$ split therefore gives the lowest FID among the tested allocations. Allocating more depth to the Semantic flow helps up to this point, but reducing Pixel flow depth further reverses the gain.

Pixel flow time shift. Figure 7 varies the time-grid parameter μ_x in Equation 14, keeping the checkpoint and step counts fixed. Uniform sampling ($\mu_x = 1$) gives FID 5.71. Shifts of 4 and 8 reduce FID to 4.27 and 3.90, with the lowest tested value, 3.87, at $\mu_x = 10$. Larger shifts of 12 and 15 give 3.91 and 4.00, respectively. These measurements motivate the selected shift of 10 for the fixed 40+10-step sampling budget.

5 Limitations

Our evaluation focuses on class-conditional ImageNet at 256×256; generalization to higher resolutions and text-conditioned generation remains to be established. Training relies on a pretrained visual encoder to define semantic targets, and sensitivity to the choice of encoder has not been systematically studied. Sampling requires sequential execution of the Semantic flow and Pixel flow, and the trade-off between inference latency and image quality requires further evaluation.

6 Conclusion

We introduced SemPix, which generates dense semantic states in a pretrained representation space and uses them to condition native pixel synthesis. Coupled training lets image-generation gradients refine semantic predictions while representation supervision retains a fixed target space. On class-conditional ImageNet 256×256, SemPix-L/16 reaches unguided FID 3.87 after 40 epochs. The L/16 and H/16 models achieve guided FIDs of 2.17 and 1.83 after 80 and 160 epochs, respectively, using 60 nominal network evaluations. Controlled ablations support the contribution of image-gradient feedback, IT initialization, and condition augmentation under the tested settings. These findings support separating semantic-state generation from pixel synthesis while coupling their learning objectives.

AI USE STATEMENT

AI tools were not used for data processing. Generative AI tools assisted with modifications to parts of the experimental code. The authors have reviewed and verified all AI-assisted work and take full responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe our method and experimental setup. Additional implementation details, training hyperparameters, sampling settings, and baseline comparison protocols are provided in Appendix A.

REFERENCES

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

Fan Bao, Chongxuan Li, Jiacheng Sun, and Jun Zhu. Why are conditional generative models better than unconditional ones? *arXiv preprint arXiv:2212.00362*, 2022. URL https://arxiv.org/abs/2212.00362.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A ViT backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=urfWb7VjmL.

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*, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/papers/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.pdf.

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

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9550–9575, 2024. URL https://proceedings.mlr.press/v235/crowson24a.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Zixuan Fu, Chong Wang, Lanqing Guo, Kailai Zhou, Jiahao Nie, and Bihan Wen. A frozen pixel-space diffusion model can guide itself with its own samples. *arXiv preprint arXiv:2607.29122*, 2026. URL https://arxiv.org/abs/2607.29122.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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. *arXiv preprint arXiv:2606.27760*, 2026. URL https://arxiv.org/abs/2606.27760.

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 *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. URL https://jmlr.org/papers/v23/21-0635.html.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13213–13232, 2023. URL https://proceedings.mlr.press/v202/hoogeboom23a.html.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18062–18071, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hoogeboom_Simpler_Diffusion_1.5_FID_on_ImageNet512_with_Pixel-space_Diffusion_CVPR_2025_paper.html.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper/2022/hash/a98846e9d9cc01cfb87eb694d946ce6b-Abstract-Conference.html.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.16064.

Theodoros Kouzelis, Spyros Gidaris, and Nikos Komodakis. Coevolving representations in joint image-feature diffusion. *arXiv preprint arXiv:2604.17492*, 2026. URL https://arxiv.org/abs/2604.17492.

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

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2504.10483.

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

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e304d374c85e385eb217ed4a025b6b63-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yifan Lu, Qi Wu, Jay Zhangjie Wu, Zian Wang, Huan Ling, Sanja Fidler, and Xuanchi Ren. PiD: Fast and high-resolution latent decoding with pixel diffusion. *arXiv preprint arXiv:2605.23902*, 2026.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40, 2024. doi: 10.1007/978-3-031-72980-5_2. URL https://arxiv.org/abs/2401.08740.

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.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5188–5196, 2015.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2304.07193.

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. *arXiv preprint arXiv:2512.04926*, 2025. URL https://arxiv.org/abs/2512.04926.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2337–2346, 2019. doi: 10.1109/CVPR.2019.00244. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Park_Semantic_Image_Synthesis_With_Spatially-Adaptive_Normalization_CVPR_2019_paper.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Giorgos Petsangourakis, Christos Sgouropoulos, Bill Psomas, Theodoros Giannakopoulos, Giorgos Sfikas, and Ioannis Kakogeorgiou. REGLUE your latents with global and local semantics for entangled diffusion. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2512.16636.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Preechakul_Diffusion_Autoencoders_Toward_a_Meaningful_and_Decodable_Representation_CVPR_2022_paper.html.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. URL https://arxiv.org/abs/2204.06125.

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*, 2022.

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

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

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? *arXiv preprint arXiv:2512.10794*, 2025. URL https://arxiv.org/abs/2512.10794.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

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

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025b.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *Advances in Neural Information Processing Systems*, volume 38, 2025c. URL https://papers.nips.cc/paper_files/paper/2025/hash/c8033b6ed6dad2af747fc8fb48525996-Abstract-Conference.html.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/0b99315234cc95e6ef281f9155b68832-Abstract-Conference.html.

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 *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d9e42b4d7163931f3689d6d6fbaa11d0-Abstract-Conference.html.

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.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

A Additional Details

A.1 Training Implementation

Optimization and augmentation. We use AdamW with learning rate 10^{-4}, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.95)$, and no weight decay. Training uses bfloat16 precision, gradient clipping at 1.0, and EMA decay 0.9999. The batch size is 64 per accelerator process. Class labels and semantic conditions are independently dropped with probability 0.1, using a null label or learned null semantic embedding. Encoder-condition noise augmentation follows Sec. 4.4 with $\rho \sim \mathcal{U}(0, 0.8)$, except for the clean-condition ablation. Flow times follow logit-normal distributions, with training time shifts of 8 and 1 for the Semantic flow and Pixel flow, respectively.

A.2 Published Baselines and Sampling Accounting

Fixed-budget comparison. Published measurements in Table 1 come from Ma et al. (2026), except PixelDDT (called PixDDT in Ma et al., 2025), CoReDi, and iREPA. The baselines reproduced by PixelGen use REPA supervision except PixelFlow-L/4. Their samplers use 50 Euler steps, except PixelFlow's 100 steps, as specified in PixelGen v2 Appendix B.1; its Table 1 caption instead states 50 steps for all methods. CoReDi-L/16 is taken from Kouzelis et al. (2026), Table 4: 200K updates with global batch size 256, no CFG, and 50 Heun steps. Its IS at 200K is not reported and is marked with a dash. The 459M JiT+iREPA result is taken from Singh et al. (2025), Appendix M.8, Table 10: 200K updates with global batch size 256, no CFG, and 50 Euler steps. We label this JiT-L/16 + iREPA according to its reported 459M model size; the source table abbreviates it as JiT+iREPA. Its reported IS of 50.72 is rounded to 50.7 for consistency with the table. We additionally evaluate JiT-L/16 without REPA using a 40-epoch checkpoint (200,200 updates), 50,000 generated images, and no guidance. It uses its native 50-step sampler: 49 Heun updates followed by one Euler update, or 99 network evaluations. Its IS is 30.0. The table reports solver updates rather than network evaluations, and parameter counts in millions.

Latent baselines. The latent rows of Table 2 use MaskDiT v2 Table 1 (Zheng et al., 2024), DiT v2 Table 2 (Peebles & Xie, 2023), SiT v2 Table 7 (Ma et al., 2024), the original MDT results in arXiv v2 Table III (Gao et al., 2023), and REPA v4 Tables 4 and 10 (Yu et al., 2025a). The U-ViT-L row uses v4 Table 7 (Bao et al., 2023), with 287M generator parameters. The DDT-XL/2 row uses the 675M, 80-epoch result in DDT v2 Table 1 (Wang et al., 2025b). For SiT, we retain the original paper's IS of 277.50. The guided 800-epoch REPA row follows the operating point in Table 10. MaskDiT's 2.075M iterations include 2M training iterations and 75K unmasking-tuning iterations. These budgets exclude autoencoder and representation-encoder pretraining.

Unguided FID and IS follow the same model size and training budget where reported: DiT Table 2, SiT Table 8 (SDE: 8.26/131.65), MDT Table III, REPA Tables 3 and 8, and DDT Table 1. REPA

reports only FID at 200 epochs, so its unguided IS remains unavailable. U-ViT's published unguided result is for H/2 rather than L/2 and is not substituted here. MaskDiT is an exception to checkpoint pairing: its unguided result uses 37.5K cosine-ratio unmasking updates, whereas its guided result uses 75K zero-ratio updates after the same 2M-step pretraining.

Parameter accounting. The second term in Params counts the additional pretrained VAE image-decoding path, including the post-quantization convolution and excluding the VAE encoder. The Stable Diffusion VAE (Rombach et al., 2022) used by these latent baselines has 49.5M such parameters, calculated from its public configuration and decoder architecture. EMA and MSE variants share this architecture. DiT, MDT, REPA, and DDT use the EMA decoder for the selected results; MaskDiT uses the MSE variant, and U-ViT-L uses the original decoder weights with the same architecture. MaskDiT's 730.1M generator count is calculated from its released architecture, including its internal transformer decoder. MDT's 675.8M follows its reported inference-model count. Training-only encoders and auxiliary heads are excluded from generator counts. SemPix-L/16 has 468M parameters across the Semantic flow and Pixel flow. The guided REPA configuration in Table 2 retains the REPA head for semantic prediction, bringing the count to 471M. This head is included because it is used during generation. SemPix-H/16 has 970M generator parameters across its two flows.

Pixel baselines. Table 2 uses PixNerd v2 Table 2 (Wang et al., 2025a), JiT v1 Tables 6, 7, 9, and 13 (Li & He, 2025), and PixelREPA v1 Table 1 and its implementation details (Shin et al., 2026). The checked PixelFlow, DeCo, and PixelGen sources do not supply full-training L-size results corresponding to their headline XL comparisons. Their 200K-step L-size evaluations belong in Table 1. A dash denotes an unavailable source result. Model-size names do not imply identical parameter counts or training costs. PixelFlow-L's parameter count follows the reproduction in Ma et al. (2026).

The H-size variants use JiT-H/16 from the same JiT tables and PixelREPA-H/16 from PixelREPA Table 1. Both use 600 training epochs and have 953M parameters. PixelU-H/16 is taken from PixelU v1 Table 4 (Guo et al., 2026), using the 600-epoch result and its reported 1168M parameter count.

For unguided JiT-L/16 and JiT-H/16, we use the original JiT baseline results reported in Table 3 of Fu et al. (2026): FID/IS of 13.85/104.2 and 7.15/151.7, respectively, at 600 training epochs. These rows use no self-guidance adapters. Guided JiT results follow Li & He (2025). Matching unguided results were not found for PixNerd-L, PixelREPA-L/H, or PixelU-H and are left as dashes. LF-DiT-L uses Table 11 of Baade et al. (2026): 465M parameters, 200 epochs, unguided FID 7.2 and guided FID 2.48; IS is not reported. Its guided result combines interval CFG for semantic timesteps with AutoGuidance for pixel timesteps, as specified in its Section 5.4. The unguided SemPix-L/16 result at 80 epochs is FID 3.27 and IS 202.9; SemPix-H/16 achieves FID 2.40 and IS 225.2 at 160 epochs.

Latent sampling. Table 2 lists each sampler followed by its reported step budget: E denotes Euler, H Heun, D DDPM, EM Euler–Maruyama, and DP DPM-Solver. Total NFE includes the conditional and unconditional CFG branches, counting them separately even when batched together, and excludes the image decoder. Before the CFG multiplier, we count two evaluations per Heun step and one per Euler, DDPM, or Euler–Maruyama step, without subtracting savings from endpoint updates. A dagger marks the nominal full-CFG upper bound for interval-guided rows; skipping unused unconditional branches can reduce their executed counts. Table 4 lists guidance settings. Full denotes guidance across the sampling trajectory; intervals retain each source's time convention. DiT and MDT use 250 DDPM steps; SiT and REPA use 250 Euler–Maruyama steps. MaskDiT uses 40 Heun steps, reported here as $40 \times 2 \times 2 = 160$ nominal evaluations including CFG; its paper reports 79 before CFG after accounting for the endpoint update, corresponding to 158 with CFG. MDT uses a power-cosine guidance schedule with maximum scale 3.8 and exponent 4. The 200-epoch REPA row uses scale 1.35; the 800-epoch row uses scale 1.8 over $[0, 0.7]$. DDT-XL uses 250 Euler steps and scale 2.0 over $[0.3, 1]$. U-ViT-L uses DPM-Solver with a budget of 50 function evaluations before CFG, or 100 total NFE; DP/50 counts internal solver stages rather than outer updates. Its released implementation uses $\epsilon_{\text{cond}} + 0.4(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}})$, equivalent to CFG scale 1.4 under the convention used in Table 4.

Table 4: Guidance configurations for Table 2. For SemPix, paired scales and intervals use Semantic / Pixel order. Semantic internal guidance uses one forward pass per update.

	Method	Epochs	Guidance scale	Interval
Latent	U-ViT-L/2	≈ 240	1.4	Full
	MaskDiT	-	1.5	Full
	DiT-XL/2	≈ 1400	1.5	Full
	SiT-XL/2	≈ 1400	1.5	Full
	MDT-XL/2	≈ 1300	Power-cosine	Full
	SiT-XL/2 + REPA	200	1.35	Full
	SiT-XL/2 + REPA	800	1.8	$[0, 0.7]$
	DDT-XL/2	80	2.0	$[0.3, 1]$
Pixel	PixNerd-L/16	160	3.5	$[0.1, 1]$
	JiT-L/16	600	2.4	$[0.1, 1]$
	JiT-H/16	600	2.2	$[0.1, 1]$
	PixelREPA-L/16	600	2.8	$[0.1, 1]$
	PixelREPA-H/16	600	2.4	$[0.1, 1]$
	PixelU-H/16	600	2.2	$[0.1, 1]$
	LF-DIT-L	200	-	-
	SemPix-L/16	80	1.75 / 1.35	$[0, 0.9]$ / $[0.1, 0.9]$
	SemPix-H/16	160	1.35 / 1.65	$[0, 0.9]$ / $[0.1, 0.9]$

Training units and guidance. For the training-progress curves in Figure 4, SemPix, JiT, JiT + REPA, and DeCo use reported epoch–FID measurements. All plotted results are without guidance. Lines connect available observations without fitting a continuous trend; the horizontal axis measures image exposure rather than compute.

Training budgets in Table 2 are expressed in epochs. For iteration-based reports, we compute image-exposure equivalents as $E = \sum_j I_j B_j / N$, where I_j and B_j are the iteration count and global batch size of training stage j, and $N = 1{,}281{,}167$. Converted values are rounded to the nearest 10 epochs and marked with $\approx$. U-ViT-L uses 300K iterations with batch size 1024, giving ≈ 240 epochs (Bao et al., 2023). DiT-XL and SiT-XL each use 7M iterations with batch size 256, giving ≈ 1400 epochs (Peebles & Xie, 2023; Ma et al., 2024). Original MDT-XL uses 6.5M iterations with batch size 256, giving ≈ 1300 epochs (Gao et al., 2023). MaskDiT's paper gives a default batch size of 1024, but its released unmasking configuration uses 64 per GPU (512 on the reported eight GPUs), compared with 128 per GPU in pretraining. The global batch of the published 75K-step tuning run is therefore unresolved; we leave its epoch total as a dash. These budgets exclude pretrained image and representation encoders and do not equate training compute. The approximately 40-epoch comparison in Table 1 corresponds to 200K updates at global batch size 256.

JiT and PixelREPA use a global batch size of 1024. Guidance scales of 2.4 for JiT-L and 2.8 for PixelREPA-L are supplied by the released JiT and PixelREPA evaluation recipes; the original JiT experiments used a JAX implementation. The corresponding H-size recipes specify 2.2 for JiT and 2.4 for PixelREPA. The PixelU recipe specifies 2.2 for H/16, with 50 Heun steps documented in the paper. PixNerd reports scale 3.5. These methods use a guidance interval $[0.1, 1]$. Interval endpoints and whether unused unconditional branches are skipped depend on implementation.

Steps, evaluations, and compute. Table 2 reports nominal sampling budgets, not measured network-call counts. The released JiT and PixelREPA samplers perform 49 Heun steps followed by one Euler step, giving $2 \times 49 + 1 = 99$ evaluations. Their released implementations evaluate both conditional and unconditional branches at every evaluation, yielding 198 per-sample network evaluations even outside the guidance interval. These are code-derived counts, not measurements of the original published runs. Under our uniform nominal convention, their 50-step Heun configurations are both listed as $50 \times 2 \times 2 = 200$ NFE including CFG. PixelU Table 4 reports the same nominal NFE as $100 \times 2 = 200$, including CFG. This convention avoids mixing endpoint-adjusted counts with nominal budgets. For the approximately 40-epoch fixed-budget comparison in Table 1, the SemPix sampler in Sec. 3.3 uses 40 Semantic flow steps and 10 Pixel flow steps without guidance. The nominal single-branch Euler counts are therefore 40 Semantic flow and 10 Pixel flow

evaluations. When enabled, internal Semantic flow guidance uses two predictions from the same forward pass and adds no second Semantic flow branch. The guided SemPix-L/16 and SemPix-H/16 results in Table 2 use 80- and 160-epoch checkpoints, respectively. Both use Euler integration with 40 Semantic flow updates and one branch each, followed by 10 Pixel flow updates. Counting both Pixel CFG branches at every update gives 50 solver updates and

$$\mathrm{NFE}_{\mathrm{SemPix}}^{\mathrm{nominal}} = 40 \times 1 + 10 \times 2 = 60. \tag{11}$$

No Heun multiplier applies. Guidance scales and intervals are listed in Table 4. For both SemPix sizes, 60 is a nominal upper bound; skipping the Pixel reference branch outside its guidance interval can reduce the executed count. Semantic internal guidance adds no branch, regardless of its interval. For per-evaluation costs C_z and C_x, the unguided and full-guidance budgets are $40C_z + 10C_x$ and $40C_z + 20C_x$, respectively, excluding fixed overheads. These are protocol-based budgets, not profiler measurements. LF-DiT-L's hybrid CFG/AutoGuidance protocol is not assigned a total NFE in this table. Summing heterogeneous network calls does not establish a speedup. Efficiency claims require measured FLOPs or latency under the same hardware, precision, batch size, and quality target.

A.3 Flow Architectures and Auxiliary Supervision

Semantic flow and main prediction. We implement F_θ with a Decoupled Diffusion Transformer (DDT) (Wang et al., 2025b), following RAE (Zheng et al., 2025). A deep trainable transformer backbone processes noisy feature tokens under time and class conditioning. A shallow diffusion head uses this context and the noisy tokens to predict the clean map $\widehat{z}$. Together they form F_θ; its main output is the predicted condition in Eq. 3. Both operate on the full $H_z \times W_z$ grid, preserving the spatial organization used for pixel conditioning.

Intermediate supervision and guidance. The intermediate prediction defined in Eq. 5 also provides the reference for internal guidance when it is enabled (Eq. 9). Configurations without intermediate supervision set $\lambda_{\mathrm{mid}} = 0$ (Sec. 4.1).

Spatially conditioned Pixel flow. The dense semantic state carries location-dependent information. We retain this correspondence by modulating pixel tokens with spatially aligned semantic features in every transformer block. Non-overlapping pixel patches are embedded into tokens with positional embeddings. Let $\mathcal{R}$ resize the semantic map to the pixel-token grid when necessary and $\mathcal{P}_s$ project its channels to the transformer width. At location i, we combine the semantic feature with time and class embeddings:

$$c_i = \mathrm{SiLU}\big(\big[\mathcal{P}_s(\mathcal{R}(s))\big]_i + e_t(t_x) + e_y(y)\big)\,. \tag{12}$$

The time and class embeddings are shared across locations; the semantic term varies spatially. In block ℓ, functions of c_i produce the shift, scale, and residual gate for each sublayer $r \in \{\mathrm{attn}, \mathrm{mlp}\}$:

$$\mathrm{adaLN}_{\ell,r}(h_i \mid c_i) = [1 + \gamma_{\ell,r}(c_i)] \odot \mathrm{Norm}(h_i) + \beta_{\ell,r}(c_i), \tag{13}$$

where h_i is the pixel token at location i, and $g_{\ell,r}(c_i)$ gates the corresponding residual update. Each block receives local semantic information, while self-attention exchanges information across locations. A final projection maps tokens back to pixel patches. The flows have separate parameters, allowing their depths and sampling budgets to be chosen independently.

A.4 Pixel Flow Time Grid

We control where the N_x Pixel flow steps fall along the trajectory using the rational time reparameterization of rectified-flow sampling (Esser et al., 2024). With uniform solver coordinates $u_k = k/N_x$, the time grid is

$$t_k = \Gamma_{\mu_x}(u_k) = \frac{u_k}{u_k + (1 - u_k)\mu_x}, \quad k = 0, \ldots, N_x, \tag{14}$$

and the flow is evaluated at $k = 0, \ldots, N_x - 1$. The positive parameter μ_x controls the allocation: $\mu_x = 1$ gives uniform steps, whereas $\mu_x > 1$ places more evaluations in the high-noise portion and fewer near the clean endpoint. The fixed-step sensitivity experiment is reported in Figure 7. Step budgets and the Pixel flow shift are specified in Sec. 4.1.