SEMANTIC TOKENS AS THE CONTEXT OF PIXEL SPACE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Context is central to how large language models generate: what has been produced earlier conditions everything produced later, and this accumulated context is what allows them to solve difficult tasks reliably. Pixel-space diffusion models, in contrast, carry almost no explicit context. The only state passed from one denoising step to the next is the noisy image itself, so any higher-level understanding of what is being generated must be re-derived from pixels at every step. This raises a natural question of what should serve as context for pixel generation. We find that coarse, locally aggregated semantic tokens make an effective context for pixel generation. Unlike a text prompt, this image-specific context is not available at the start of generation. We therefore propose **SemPix**, which jointly generates semantic tokens and pixels from noise through flow matching and carries the semantic state across denoising steps, allowing the evolving context to inform later pixel predictions. Pixel and semantic tokens interact through joint self-attention in a shared Transformer backbone, with separate heads mapping them to their respective targets. Experiments show that this generated context substantially improves pixel generation, and that both its form and its evolution matter. Locally aggregated features are more effective than dense or global alternatives, while the context is most useful when it persists across denoising steps and directly informs pixel prediction. On class-conditional ImageNet-256, SemPix-XL reaches 4.93 unguided FID after 40 epochs and 1.72 guided FID after 160 epochs. These results suggest that, as in language modeling, what a pixel generator carries along its trajectory matters as much as how it predicts the image.

1 INTRODUCTION

Context plays an important role in generative modeling: information accumulated earlier in generation can shape what is produced later. Autoregressive language models make this role explicit by carrying previously generated tokens forward as context for subsequent predictions. Pixel-space generative models organize generation differently. The evolving pixels are typically the primary sample-specific state carried from one denoising step to the next, while higher-level information about the emerging image remains implicit in them. This motivates a basic question for pixel generation: **what should serve as context along the generation trajectory?**

Pretrained visual representations already provide strong signals for learning pixel generators. Representation-alignment objectives encourage intermediate generator features to match pretrained visual features, while perceptual objectives constrain predicted images in feature space Yu et al. (2025); Ma et al. (2026b). These approaches demonstrate the value of visual semantics during training. The visual representation itself, however, is typically not maintained as a sample-specific state from one sampling step to the next. We ask whether semantic information can play a different role: not only supervising generation, but also serving as context that evolves with the image being generated.

Such context poses a particular challenge in pixel generation. A text prompt or class label can be supplied before sampling begins, but the semantics of the particular image that will emerge from noise are not available in advance. A natural alternative is therefore to generate this context together with the pixels. As generation proceeds, the semantic representation can evolve with the image and persist across denoising steps, making the evolving semantic state available to later pixel predictions.

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.

Rather than treating the visual representation as a fixed auxiliary stream, we ask what representation should serve as the generated state itself, and what properties make that state useful as context across the generation trajectory.

What form should this generated context take? Dense visual features preserve rich local information, but closely reproduce the detailed visual representation already available through the pixels. A single global token lies at the opposite extreme, compressing away much of the spatial organization of the image. We instead consider a representation that is more abstract than dense patch features while remaining spatially structured. Locally aggregating visual features provides such a middle ground, producing a coarse set of semantic tokens that can describe the emerging image without reproducing its full dense visual representation.

We introduce **SemPix**, which uses these semantic tokens as generated context for pixel-space generation. SemPix jointly generates pixels and semantic tokens from noise through flow matching and carries the semantic state across denoising steps. Pixel and semantic tokens interact through joint self-attention in a shared Transformer backbone, while separate prediction heads map them to their respective targets. During training, semantic targets are constructed by locally aggregating features from a frozen visual encoder. During sampling, no visual encoder is required: the semantic state evolves together with the pixels and directly informs subsequent pixel predictions.

Our experiments show that both the **form and evolution** of this generated context matter. Locally aggregated semantic features are more effective than dense or global alternatives. Importantly, their advantage is not explained simply by reducing the number of tokens: aggregated content remains effective at matched sequence length and across different integration mechanisms. The context is also most effective when it persists across the trajectory and can directly inform pixel prediction. Variants with no semantic carryover or blocked semantic-to-pixel interaction perform substantially worse. These controls indicate that the benefit comes not merely from adding another semantic objective, but from making semantic information part of the state carried through generation.

On class-conditional ImageNet-256, SemPix-XL reaches **4.93 unguided FID after 40 epochs** and **1.72 guided FID after 160 epochs**. Together with the controlled studies above, these results suggest that pixel-space generation is shaped not only by how pixels are predicted, but also by what context is carried along the generation trajectory.

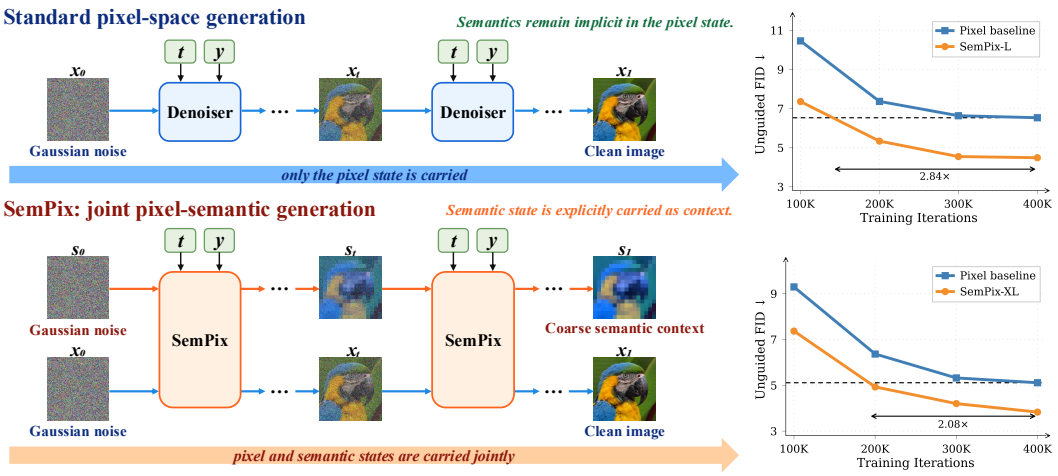
Our main contributions are:

- We propose SemPix, which treats semantic tokens as generated context for pixel-space generation and jointly evolves this context with the pixels.
- We show that coarse, locally aggregated visual features provide more effective context than dense or global alternatives, with the benefit persisting at fixed sequence length and across different integration mechanisms.
- We show that semantic context is most useful when it persists across denoising steps and directly informs pixel prediction, leading to strong generation quality on ImageNet-256.

2 Related Work

Pixel-space generation. Pixel-space generative models synthesize images directly in image space without relying on a pretrained image autoencoder. Early diffusion and score-based models operate directly on pixels Ho et al. (2020); Song & Ermon (2019); Song et al. (2021); Dhariwal & Nichol (2021). Recent work has revisited direct pixel generation with designs that improve scalability and efficiency at high resolution, including resolution-aware diffusion, cascaded flow modeling, neural-field decoding, and clean-image prediction Hoogeboom et al. (2023); Chen et al. (2025); Wang et al. (2026a); Li & He (2026). Other approaches restructure pixel modeling by separating global structure from local detail or low- from high-frequency content Yu et al. (2026); Chen et al. (2026); Ma et al. (2026a), while EPG uses self-supervised pretraining followed by end-to-end pixel-space generative training Lei et al. (2026). These advances substantially strengthen direct pixel generation, while the evolving image remains the primary sample-specific state carried along the generation trajectory.

Visual representation supervision. Pretrained visual representations provide effective supervisory signals for generative modeling. REPA aligns intermediate denoiser features with frozen visual



(a) Pixel and semantic trajectories. (b) Training convergence.

Figure 1: **Carrying semantic context through pixel generation.** (a) Standard pixel-space generation propagates the evolving pixel state across denoising steps. SemPix additionally maintains an image-specific semantic state that is generated from noise, jointly evolved with the pixels, and reused as context for subsequent pixel predictions. (b) Training convergence on ImageNet-256 without guidance. SemPix improves over the pixel-only baseline at both model scales.

representations Yu et al. (2025), and REPA-E extends representation alignment to end-to-end tuning of the image tokenizer and diffusion model Leng et al. (2025). Recent analyses further study which properties of representation targets matter for generation and how alignment behaves in pixel-space models: iREPA highlights the importance of spatial structure Singh et al. (2026), while PixelREPA adapts representation alignment to JiT through transformed alignment targets and masked adaptation Shin et al. (2026). Beyond intermediate feature alignment, PixelGen applies perceptual supervision directly to predicted images using complementary LPIPS and DINO losses Ma et al. (2026b). Together, these works show that pretrained visual representations can provide strong semantic supervision for learning image generators.

Representation states in generation. Visual representations can play a role beyond training-time supervision by becoming explicit variables in the generative process. ReDi jointly generates dense visual features with image latents Kouzelis et al. (2025), while REG augments latent-space generation with a single global visual token Wu et al. (2025). Semantic-First Diffusion separates semantic and texture representations and organizes their evolution with asynchronous denoising Pan et al. (2026), while Latent Forcing introduces an auxiliary latent trajectory to order pixel generation Baade et al. (2026). V-Co jointly denoises pixels and dense visual features and systematically studies design choices for pixel-space visual co-denoising Lin et al. (2026). Our SemPix focuses on the semantic state itself: what representation should be carried as context, and how that state should participate in the generation trajectory to support pixel prediction. Our experiments show that compact spatial semantic tokens improve pixel generation when they persist across denoising steps and directly inform pixel prediction.

3 Method

Figures 1 and 2 illustrate how **SemPix** turns compact visual representations into an evolving state for pixel generation. Standard pixel-space generation propagates the evolving pixel state across denoising steps. SemPix instead concatenates noisy pixel and semantic tokens at the input, predicts both clean targets, and jointly updates their states. The semantic state is carried forward to guide subsequent pixel predictions. A shared Transformer backbone processes the concatenated pixel and semantic tokens through joint self-attention, and two separate context-modulated refinement heads predict their respective targets.

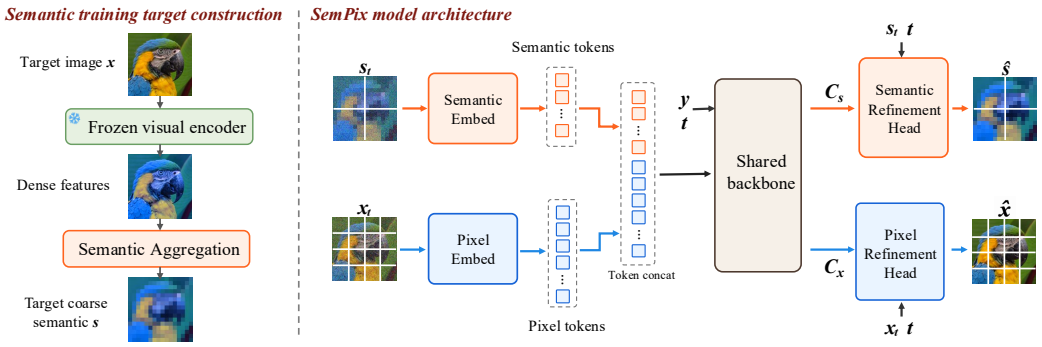


Figure 2: **SemPix framework.** A frozen DINOv2 encoder provides compact spatial semantic targets during training. Noisy pixel and semantic tokens are concatenated into a joint sequence and processed by a shared Transformer backbone conditioned on timestep and class. Separate refinement heads combine the shared context with the corresponding noisy input embeddings to predict clean pixel and semantic targets. The visual encoder is used only during training. At inference, both states are generated from noise and evolved jointly.

3.1 Joint Flow Matching for Pixels and Semantics

Let $x \in \mathbb{R}^{3\times H\times W}$ be an image, $s \in \mathbb{R}^{N_{\text{sem}}\times C_{\text{sem}}}$ its compact semantic representation, and c the class condition. We model the joint distribution of images and their semantic representations with flow matching. At time $t \in [0,1]$, we independently corrupt the two targets:

$$x_t = tx + (1-t)\epsilon_x, \qquad s_t = ts + (1-t)\epsilon_s, \tag{1}$$

where ϵ_x and ϵ_s are standard Gaussian noise with the respective target shapes. The trajectory runs from noise at $t=0$ to clean data at $t=1$.

Following the clean-image prediction formulation of JiT Li & He (2026), SemPix predicts both clean targets:

$$(\hat{x}, \hat{s}) = f_\theta(x_t, s_t, t, c), \qquad \hat{v}_x = \frac{\hat{x}-x_t}{1-t}, \quad \hat{v}_s = \frac{\hat{s}-s_t}{1-t}. \tag{2}$$

The clean-target estimates $\hat{x}$ and $\hat{s}$ define velocities for a coupled generation process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_t \\ s_t \end{pmatrix} = \begin{pmatrix} \hat{v}_x(x_t, s_t, t, c) \\ \hat{v}_s(x_t, s_t, t, c) \end{pmatrix}. \tag{3}$$

Each pixel prediction can therefore use the current semantic state, while each semantic prediction incorporates the evolving image. Integrating both velocities preserves this interaction across sampling steps.

The target velocities associated with Eq. (1) are $v_x = x - \epsilon_x$ and $v_s = s - \epsilon_s$. We use the standard pixel velocity loss and a temporally weighted semantic loss:

$$\begin{aligned} \mathcal{L}_{\text{img}} &= \mathbb{E}\big[\|\hat{v}_x - v_x\|_2^2\big] = \mathbb{E}\left[\frac{\|\hat{x}-x\|_2^2}{(1-t)^2}\right], \\ \mathcal{L}_{\text{sem}} &= \mathbb{E}\big[(1-t)\|\hat{v}_s - v_s\|_2^2\big] = \mathbb{E}\left[\frac{\|\hat{s}-s\|_2^2}{1-t}\right]. \end{aligned} \tag{4}$$

In clean-target form, the semantic loss uses a $(1-t)^{-1}$ weight, while the pixel loss uses $(1-t)^{-2}$. The semantic loss is averaged over token positions and feature channels. Appendix B.7 evaluates the semantic prediction targets and weighting choices.

3.2 Compact Semantic Context

We construct semantic targets by locally pooling patch features from a frozen DINOv2 encoder Oquab et al. (2024). We use average pooling to compute the coarse semantic tokens for each neighborhood of the feature grid.

Table 1: **Controlled comparison under a matched training budget.** We compare generators on ImageNet-256 without classifier-free guidance. All models are evaluated with 50-step Euler sampling.

Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
DiT-L/2	41.93	36.5	0.52	0.59
REPA-L/2	16.14	87.3	0.65	0.63
PixNerd-L/16	37.49	43.0	0.46	0.62
PixelFlow-L/4	54.33	24.7	0.43	0.58
DeCo-L/16	31.35	48.4	0.51	**0.65**
JiT-L/16	23.64	63.4	0.56	0.63
PixelGen-L/16	7.53	131.7	0.72	0.60
SemPix-L/16	**5.33**	**157.2**	**0.75**	0.59

Table 2: **Unguided generation and training convergence on ImageNet-256.** We report representative direct pixel-space generators together with multiple SemPix-XL checkpoints to expose the training trajectory.

Method	Epochs	FID ↓	IS ↑	Prec. ↑	Rec. ↑
ADM	400	10.94	—	0.69	0.63
PixelFlow-XL/4	320	12.23	103.3	0.63	0.66
PixNerd-XL/16	320	15.61	88.9	0.59	**0.68**
DeCo-XL/16	320	14.88	88.2	0.60	**0.68**
PixelGen-XL/16	80	5.11	159.2	0.72	0.63
SemPix-XL/16	40	4.93	154.7	0.73	0.60
SemPix-XL/16	80	3.83	171.4	**0.74**	0.63
SemPix-XL/16	160	**3.35**	**174.8**	**0.74**	0.64

The noisy semantic tokens are concatenated with pixel tokens at the input and participate throughout the shared backbone. The class condition c is retained, and the joint sequence contains $N_{\text{pix}} + N_{\text{sem}}$ tokens. Local aggregation preserves spatial organization in a small semantic state that evolves alongside pixels. Appendix A.4 gives the feature extraction and preprocessing details.

3.3 Dual-Side Refinement

Figure 2 shows how SemPix combines a shared Transformer backbone with a *Pixel Refinement Head* and a *Semantic Refinement Head*. Joint attention allows pixel and semantic tokens to exchange information, while their predictions target pixel values and visual features, respectively. The bypass connections provide each head with its corresponding input embeddings directly. The shared backbone outputs condition the heads through token-wise modulation. Appendix A.2 provides the formulation and implementation details.

3.4 Sampling

Figure 1(a) illustrates how SemPix maintains both pixel and semantic states throughout sampling: for solver times $t_0 = 0 < \cdots < t_K < 1$, let $(x_k, s_k) = (x_{t_k}, s_{t_k})$. We initialize (x_0, s_0) with independent Gaussian noise and integrate Eq. (3). An Euler step takes the form

$$x_{k+1} = x_k + \Delta t_k \hat{v}_{x,k}, \qquad s_{k+1} = s_k + \Delta t_k \hat{v}_{s,k}, \tag{5}$$

where $\Delta t_k = t_{k+1} - t_k$ and both velocities are evaluated from (x_k, s_k, t_k, c). The updated semantic state s_{k+1} enters the next model evaluation together with x_{k+1}, providing a continuous semantic trajectory for image generation. Sampling configuration details are provided in Appendix A.5.

4 Experiments

We evaluate SemPix on ImageNet-256. We first report generation quality and training convergence, then examine how semantic tokens function as generative context and which representations make this context effective. Finally, we ablate the representation integration and architecture design.

4.1 Experimental Setup

We evaluate SemPix on class-conditional ImageNet-256. We use SemPix-XL for the main generation results, evaluating checkpoints at 200K, 400K, and 800K training steps, corresponding to approximately 40, 80, and 160 epochs. Controlled comparisons and design studies use SemPix-L trained for 200K steps. We report FID, Inception Score (IS), Precision, and Recall over 50K generated images. Unless otherwise stated, our SemPix-L design studies use both REPA and perceptual supervision. The controlled comparison therefore uses the same auxiliary-supervision recipe. This provides a common experimental setting for studying the effect of the generated semantic state and its design. SemPix-L evaluations use 50 step Euler sampling without classifier-free guidance, while the main SemPix-XL evaluations use 50 step Heun sampling. Full training and evaluation details are provided in Appendix A.1.

Table 3: **Guided ImageNet-256 generation.** We compare SemPix with representative latent-space reference models and recent direct pixel-space generators. Multiple SemPix checkpoints are included to show its scaling with training.

Method	Params	Epochs	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Latent-space reference models						
DiT-XL/2 Peebles & Xie (2023)	675+49M	1400	2.27	278.2	0.83	0.57
SiT-XL/2 Ma et al. (2024)	675+49M	1400	2.06	270.3	0.82	0.59
REPA-XL/2 Yu et al. (2025)	675+49M	800	1.42	305.7	0.80	0.65
DDT-XL/2 Wang et al. (2026b)	675+49M	400	1.26	310.6	0.79	0.65
Direct pixel-space models						
PixelFlow-XL/4 Chen et al. (2025)	677M	320	1.98	282.1	0.81	0.60
DiP-XL/16 Chen et al. (2026)	631M	600	1.79	281.9	0.80	0.63
DeCo-XL/16 Ma et al. (2026a)	682M	800	1.62	301.0	0.80	0.62
JiT-H/16 Li & He (2026)	953M	600	1.86	303.4	0.78	0.62
Latent Forcing (LF-DiT-L) Baade et al. (2026)	465M	200	2.48	—	—	—
EPG-XL/16 Lei et al. (2026)	583M	800	2.04	283.2	0.80	0.61
PixelGen-XL/16 Ma et al. (2026b)	676M	160	1.83	293.6	0.79	0.63
PixelREPA-H/16 Shin et al. (2026)	953M	600	1.81	317.2	—	—
PixelDiT-XL Yu et al. (2026)	797M	320	1.61	292.7	0.78	0.64
V-Co-L/16 Lin et al. (2026)	918M	500	1.72	245.3	—	—
SemPix-XL/16	687M	80	2.23	301.2	0.77	0.61
SemPix-XL/16	687M	160	1.72	316.1	0.78	0.62

4.2 Semantic Context Improves Pixel Generation

We first evaluate whether introducing an evolving semantic state improves pixel generation under a controlled training budget. We then examine how the benefit scales with model size and training, before placing the resulting model in the broader guided-generation landscape.

Matched-budget comparison. Table 1 compares L-sized latent-space and direct pixel-space generators trained for the same 200K-step budget. SemPix-L improves FID from 7.53 for PixelGen-L/16 to 5.33, while increasing IS from 131.7 to 157.2 and Precision from 0.72 to 0.75. This controlled comparison establishes that introducing the evolving semantic state substantially improves generation within the same training regime.

Unguided generation and convergence. Table 2 and Figure 1(b) show that the gain persists when generation is evaluated without classifier-free guidance. SemPix-XL reaches 4.93 FID after 40 epochs, already improving over the 5.11 FID of PixelGen-XL/16 trained for 80 epochs. At the same 80-epoch budget, SemPix further improves to 3.83 FID, and reaches 3.35 FID after 160 epochs. The convergence curves show the same trend at both the L and XL scales, with SemPix achieving lower unguided FID throughout the reported training checkpoints. These results show that the evolving semantic state strengthens the underlying pixel-generation process and that its benefit persists as training and model scale increase.

Guided system-level comparison. Table 3 places SemPix in the broader guided-generation landscape. SemPix-XL reaches 2.23 FID after 80 epochs and 1.72 FID after 160 epochs, with an IS of 316.1 at the final checkpoint. At 160 epochs, SemPix matches the 1.72 FID reported by V-Co-L/16 while using a 687M-parameter backbone compared with 918M parameters for V-Co and a substantially shorter reported training schedule. Together with the unguided results, this places SemPix as a competitive direct pixel-space generator with strong training convergence.

Qualitative comparison. Figure 4 compares samples from L-sized models trained for the same 200K-step budget and sampled without guidance. Relative to JiT, the perceptually supervised baseline improves local image detail, while SemPix further produces clearer object contours and fine-grained structure in the shown examples. In Figure 3 we show the SemPix-XL samples on ImageNet-256 with classifier-free guidance.



Figure 3: **SemPix-XL samples on ImageNet-256.** Class-conditional images generated with classifier-free guidance.

Table 4: **Semantic context across the generation trajectory.** We intervene on semantic-state persistence, semantic-to-pixel interaction, and auxiliary-state content under the same architecture and training setup.

Variant	FID ↓	IS ↑	Prec. ↑	Rec. ↑
No semantic carryover	8.30	133.6	0.70	**0.61**
Block semantic→pixel	8.12	131.3	0.69	**0.61**
Downsampled pixel auxiliary state	8.58	125.2	0.69	0.60
Full interaction (SemPix)	**5.33**	**157.2**	**0.75**	0.59

4.3 Evolving Semantic Context Across the Generation Trajectory

We next ask what makes the generated semantic state useful as trajectory-level context. Using the same 32-token SemPix-L setup, we compare full SemPix with three controlled variants: *no semantic carryover* resets the semantic state at every denoising step, *blocked semantic-to-pixel attention* preserves the semantic trajectory but prevents pixels from reading it, and *downsampled pixel auxiliary state* replaces semantic targets with a same-shaped RGB state.

Evolving semantic context. Full interaction achieves 5.33 FID, compared with 8.30 when the semantic state is reset at each denoising step, 8.12 when semantic-to-pixel access is blocked, and 8.58 with the pixel auxiliary state. Our results show that pixel space generation benefits from propagating the semantic state across steps and making it accessible to subsequent pixel predictions, while an additional evolving state alone is insufficient.

Complementarity with visual supervision. Table 5 shows that semantic context remains effective across increasingly strong visual-supervision settings. Adding semantic context improves FID from 36.45 to 28.47 without REPA or perceptual supervision, from 23.64 to 19.53 with REPA, and from 7.53 to 5.33 with both REPA and perceptual losses. This shows that carrying semantics as generation-time context provides a complementary benefit to representation and perceptual supervision used during training.

Instance-specific semantic context. Figure 5 compares the predicted semantic state with semantic features extracted from the corresponding predicted image. Matched pixel–semantic trajectories remain substantially more consistent than independently sampled same-class or different-class pairs, showing that the semantic state tracks the particular image emerging along the trajectory rather than only the class condition. Together, these results establish the semantic state as an image-specific context that is generated, carried forward, and reused throughout pixel generation.

4.4 Coarse Spatial Semantics Provide Effective Generative Context

We next study what form the semantic representation should take as generative context.



Figure 4: **Qualitative comparison.** Generated under the controlled 200K-step setting without classifier-free guidance.

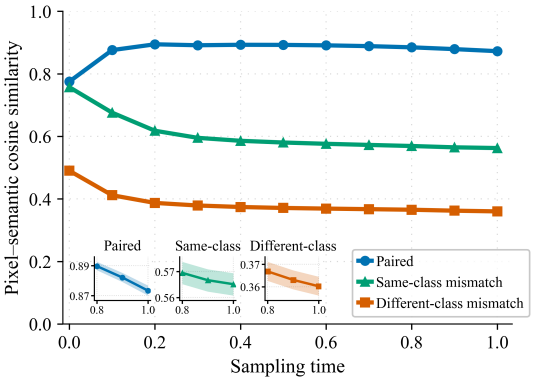


Figure 5: **Pixel–semantic consistency.** Matched trajectories are more consistent than independent same- or different-class pairs.

Table 5: **Semantic state with visual supervision.** SemPix-L at 200K steps without CFG.

REPA	Perc. Loss	Sem State	FID ↓	IS ↑	Prec. ↑	Rec. ↑
✗	✗	✗	36.45	40.9	0.48	0.63
✗	✗	✓	**28.47**	**53.1**	**0.53**	0.63
✓	✗	✗	23.64	63.4	0.56	0.63
✓	✗	✓	**19.53**	**81.0**	**0.57**	**0.64**
✓	✓	✗	7.53	131.7	0.72	**0.60**
✓	✓	✓	**5.33**	**157.2**	**0.75**	0.59

Table 6: **Granularity of semantic context.** SemPix-L at 200K steps without CFG.

Representation	Tokens	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Single global token	1	7.89	136.7	0.67	0.58
Coarse semantic	4	7.34	141.2	0.71	**0.61**
Coarse semantic	8	6.75	145.1	0.72	0.60
Coarse semantic	16	6.14	151.2	0.73	0.59
Coarse semantic	32	5.33	**157.2**	**0.75**	0.59
Coarse semantic	64	**5.22**	156.9	**0.75**	0.57
Coarse semantic	128	5.61	147.3	**0.75**	0.56
Dense semantic	256	6.74	131.7	0.74	0.55

Semantic context granularity. Table 6 shows a clear intermediate regime: a single global token and dense 256-token features obtain 7.89 and 6.74 FID, whereas coarse spatial representations reach 5.33 with 32 tokens and 5.22 with 64 tokens. This suggests that effective context should preserve spatial organization while abstracting away patch-level detail, rather than maximizing either compression or spatial granularity. We use 32 tokens by default, which provides comparable generation quality with a more compact semantic state.

Beyond sequence length and integration design. The benefit of coarse semantics is not explained by sequence shortening alone. At a matched 256-token sequence length, repeating the 32 aggregated features improves FID from 6.74 to 5.24 over dense features. The same preference for aggregated semantics holds across embedding addition, channel concatenation, and explicit semantic-token integration, indicating that the gain is not tied to a particular fusion mechanism. Appendix B.4 shows the same dense-to-coarse improvement with both DINOv2 and SigLIP. Appendix B.5 further shows that the compact 32-token state maintains its advantage over dense features through 600K training steps.

Compactness versus semantic abstraction. Taken together, these results show that sequence shortening alone does not account for the benefit of aggregation. Aggregated semantic representations remain effective at matched sequence length, across integration mechanisms and visual encoders, and over longer training schedules. We therefore use the 32-token state as a compact spatial abstraction of the emerging image rather than simply as a shorter dense feature sequence.

4.5 Architecture Design

We finally examine how to realize joint pixel–semantic modeling through backbone sharing and target-specific prediction.

Table 7: **Local aggregation at matched sequence length and across integration mechanisms.** Aggregated variants repeat 32 locally aggregated features to 256 positions when compared with dense semantic sequences, keeping the semantic sequence length fixed. SemPix uses the compact 32-token state by default.

Integration	Semantic representation	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Explicit semantic tokens	Dense semantic	6.74	131.7	0.74	0.55
	Repeated coarse semantic	**5.24**	145.2	**0.76**	0.55
	Coarse semantic	5.33	**157.2**	0.75	**0.59**
Embedding addition	Dense semantic	6.87	132.0	0.75	0.54
	Repeated coarse semantic	**5.65**	**144.3**	**0.76**	0.54
Channel concatenation	Dense semantic	6.62	138.9	0.72	**0.57**
	Repeated coarse semantic	**5.79**	**150.2**	**0.75**	0.56

Table 8: **Shared backbone and target-specific refinement.** Group A compares shared and dual-stream architectures. Group B compares prediction-head designs under the shared backbone. All variants use the same 32-token semantic representation.

Architecture	Params	FID ↓	IS ↑	Prec. ↑	Rec. ↑
A. Backbone architecture					
Parameter-matched dual stream	463.80M	6.81	141.9	0.72	0.59
Full-width dual stream	915.65M	**5.15**	**158.6**	0.74	**0.60**
Shared backbone (SemPix)	464.36M	5.33	157.2	**0.75**	0.59
B. Prediction heads					
Separate final layers	462.26M	6.53	142.6	0.73	0.58
Dual JiT heads	464.36M	6.15	143.0	0.73	0.59
Semantic-side refinement	463.31M	6.26	145.0	0.73	0.59
Pixel-side refinement	463.31M	5.83	153.4	0.74	0.59
Dual refinement heads (SemPix)	464.36M	**5.33**	**157.2**	**0.75**	0.59

Shared versus dual-stream backbones. Table 8 compares our shared backbone with dual-stream designs following V-Co Lin et al. (2026). At a matched parameter budget, SemPix achieves 5.33 FID with 464.36M parameters, compared with 6.81 for the 463.80M dual-stream variant. A full-width dual-stream model reaches 5.15 FID with 915.65M parameters. Thus, the shared backbone approaches the full-width dual-stream result with roughly half the parameters while substantially improving over the parameter-matched alternative. Appendix B.1 shows a similar compact-state trend under a parameter-matched dual-stream backbone.

Target-specific refinement. With the shared backbone fixed, separate final layers achieve 6.53 FID, while dual JiT heads, following the prediction design used in Latent Forcing Baade et al. (2026), improve this to 6.15. Refining only the semantic or pixel branch gives 6.26 and 5.83 FID, respectively, while refining both reaches 5.33 FID. These results support sharing pixel–semantic context in the backbone while retaining target-specific refinement for both prediction domains.

5 Conclusion

We introduced SemPix, which treats an image-specific semantic state as generated context for pixel-space generation. This state emerges jointly with the pixels from noise, is carried across denoising steps, and requires no visual encoder during sampling. Our experiments show that both its form and trajectory role matter: coarse, locally aggregated semantics outperform global and dense alternatives, while the state is most effective when propagated across steps and made available to subsequent pixel predictions. The generated state tracks the emerging image and complements training-time visual supervision. On class-conditional ImageNet-256, SemPix-XL reaches 4.93 unguided FID after 40 epochs and 1.72 guided FID after 160 epochs. Together, these results identify generated semantic context as an important design axis for pixel-space generation alongside how the image itself is predicted.

AI USE STATEMENT

Generative AI tools were used solely for language editing and polishing of the manuscript. They were not used to develop the research ideas or methodology, conduct experiments, analyze or interpret results, or generate any reported data. All scientific content and conclusions were produced and verified by the authors.

REPRODUCIBILITY STATEMENT

We provide detailed model, training, and evaluation configurations in Appendix A, including semantic-token construction, sampling, classifier-free guidance, and the implementation of trajectory interventions. Appendix B reports additional architectural, representation, objective, and training ablations, while Appendix C provides computational-cost measurements. Together, these details specify the settings used for the reported experiments and controlled comparisons.

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.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-Space Generative Models with Flow. arXiv preprint arXiv:2504.07963, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36136–36146, June 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021.

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, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024.

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 (ICCV)*, pp. 23164–23173, October 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

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. PMLR, 23–29 Jul 2023.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Boosting Generative Image Modeling via Joint Image-Feature Synthesis. In *Advances in Neural Information Processing Systems*, volume 38, pp. 16685–16714. Curran Associates, Inc., 2025. doi: 10.52202/085713-0563.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is No VAE: End-to-End Pixel-Space Generative Modeling via Self-Supervised Pre-Training. In *International Conference on Learning Representations*, pp. 102818–102839, 2026.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for End-to-End Tuning of Latent Diffusion Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272, October 2025.

Tianhong Li and Kaiming He. Back to Basics: Let Denoising Generative Models Denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36115–36125, June 2026.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, Yatian Pang, and Li Yuan. UniWorld-V1: High-Resolution Semantic Encoders for Unified Visual Understanding and Generation. arXiv preprint arXiv:2506.03147, 2025.

Han Lin, Xichen Pan, Zun Wang, Yue Zhang, Chu Wang, Jaemin Cho, and Mohit Bansal. V-Co: A Closer Look at Visual Representation Alignment via Co-Denoising. In *Computer Vision – ECCV 2026*, pp. 380–398. Springer Nature Switzerland, 2026. ISBN 978-3-032-37232-1. doi: 10.1007/978-3-032-37232-1_21.

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 *Computer Vision – ECCV 2024*, pp. 23–40. Springer Nature Switzerland, 2024. ISBN 978-3-031-72980-5. doi: 10.1007/978-3-031-72980-5_2.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. DeCo: Frequency-Decoupled Pixel Diffusion for End-to-End Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43600–43610, June 2026a.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. PixelGen: Improving Pixel Diffusion with Perceptual Supervision. arXiv preprint arXiv:2602.02493, 2026b.

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 Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, January 2024.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43664–43674, June 2026.

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

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 (CVPR)*, pp. 10684–10695, June 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494. Curran Associates, Inc., 2022. doi: 10.52202/068431-2643.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation Alignment for Just Image Transformers is not Easier Than You Think. In *Computer Vision – ECCV 2026*, pp. 603–620. Springer Nature Switzerland, 2026. ISBN 978-3-032-37271-0. doi: 10.1007/978-3-032-37271-0_33.

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 *International Conference on Learning Representations*, pp. 33807–33845, 2026.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the Diffusability of Autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55876–55905. PMLR, 13–19 Jul 2025.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191, 2024a.

Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Exploring DCN-like architecture for fast image generation with arbitrary resolution. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87959–87977. Curran Associates, Inc., 2024b. doi: 10.52202/079017-2792.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel Neural Field Diffusion. In *International Conference on Learning Representations*, pp. 43559–43580, 2026a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled Diffusion Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40633–40642, June 2026b.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ziyi Xia, Ze Liu, Chaofan Li, Haoge Deng, Kun Luo, Bo Zhang, Jiajun Zhang, Dong Liu, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Towards Instruction-Aligned Multimodal Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21964–21975, June 2026.

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, pp. 7714–7743. Curran Associates, Inc., 2025. doi: 10.52202/085713-0264.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One Single Transformer to Unify Multimodal Understanding and Generation. In *International Conference on Learning Representations*, pp. 28240–28264, 2025.

Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. FasterDiT: Towards Faster Diffusion Transformers Training without Architecture Modification. In *Advances in Neural Information Processing Systems*, volume 37, pp. 56166–56189. Curran Associates, Inc., 2024. doi: 10.52202/079017-1787.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712, June 2025.

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*, pp. 87400–87442, 2025.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel Diffusion Transformers for Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14273–14282, June 2026.

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 (CVPR)*, pp. 586–595, June 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion Transformers with Representation Autoencoders. In *International Conference on Learning Representations*, pp. 35791–35820, 2026.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast Training of Diffusion Models with Masked Transformers. *Transactions on Machine Learning Research*, March 2024.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model. In *International Conference on Learning Representations*, pp. 6446–6469, 2025.

A Implementation and Evaluation Details

A.1 Experimental Configurations

Models and training. We train class-conditional models at two scales: SemPix-XL for the main results and SemPix-L for comparisons and design analysis. The XL models are evaluated at 40, 80, and 160 epochs. L-sized model comparisons and ablations use 200K training steps, approximately 40 epochs. We use a global batch size of 256, AdamW with a constant learning rate of 10^{-4}, logit-normal timestep sampling, and BF16 mixed precision. SemPix builds on the JiT pixel generator and retains the REPA and perceptual supervision used by PixelGen. The semantic state contains 32 locally pooled DINOv2 tokens. Complete configurations are given in Table 9.

The higher-resolution experiment in Table 17 evaluates ImageNet-512 at 400K steps (80 epochs). The 160-epoch XL training budget corresponds to approximately 800K steps.

Evaluation. We evaluate 50K generated images using the ADM evaluation suite Dhariwal & Nichol (2021) with the standard ImageNet-256 reference batch, reporting Fréchet Inception Distance (FID), Inception Score (IS), Precision, and Recall. Main XL evaluations use 50-step Heun sampling with and without classifier-free guidance (CFG), while L-sized controlled comparisons use 50-step Euler sampling without CFG. Our controlled and unguided comparisons focus on direct pixel-space generators, isolating the setting addressed by SemPix. For the guided system-level comparison, we additionally include representative latent-space diffusion models to situate direct pixel generation in the broader generative landscape.

A.2 Dual-Side Refinement Details

Joint attention allows pixel and semantic tokens to exchange information, while their predictions target RGB values and visual features, respectively. SemPix combines a shared Transformer backbone with dedicated prediction modules for these two targets. Let $e_x = \mathcal{P}_x(x_t)$ and $e_s = \mathcal{P}_s(s_t)$ denote the pixel and semantic input embeddings. The same Transformer blocks process the concatenated sequence through joint self-attention and produce context for both token groups:

$$(h_x, h_s) = \mathcal{B}_\theta([e_x; e_s], t, c)\,, \tag{6}$$

where the brackets denote sequence concatenation, and h_x, h_s are the corresponding output token groups. Through joint attention, each group incorporates information from both noisy states.

Figure 2 shows the *Pixel Refinement Head* and *Semantic Refinement Head* for the two prediction targets. The bypass connections provide each head with its corresponding input embeddings directly. The shared backbone outputs condition the heads through token-wise modulation:

$$\begin{aligned} \hat{x} &= \mathcal{O}_x(\mathcal{D}_x(e_x; h_x, t))\,, \\ \hat{s} &= \mathcal{O}_s(\mathcal{D}_s(e_s; h_s, t))\,. \end{aligned} \tag{7}$$

Here, $\mathcal{D}_x, \mathcal{D}_s$ are separate refinement stacks, and $\mathcal{O}_x, \mathcal{O}_s$ map their outputs to RGB patches and semantic features. The pixel patches are reassembled into $\hat{x}$. This design gives each prediction access to its current noisy state and the context learned through joint interaction.

For each branch, a linear projection of the shared context is combined with the timestep embedding to form a condition at every token. Each refinement block maps this condition to scale, shift, and residual gates for its self-attention and feed-forward layers. Attention operates within the corresponding branch. SemPix-L uses 20 shared Transformer blocks and two refinement blocks per head.

A.3 Representation and Perceptual Supervision

We retain REPA Yu et al. (2025) to align pixel-token features after the eighth shared block with DINOv2 targets, using a DINOv2 model with register tokens Oquab et al. (2024) as the feature extractor. Following PixelGen Ma et al. (2026b), we also supervise the predicted image with LPIPS and a DINO perceptual loss, activated for $t > 0.3$, with weights 0.1 and 0.01, respectively. LPIPS measures similarity in VGG features, and the DINO term compares visual features of the predicted and

Table 9: Model and training configurations. Guidance settings refer to the main guided XL results; unguided evaluation uses scale 1.0 on both branches.

	SemPix-L	SemPix-XL
Architecture		
Total Transformer blocks	24	28
Shared backbone blocks	20	24
Hidden dimension	1024	1152
Attention heads	16	16
Params	464.36M	687M
Embedding bottleneck dimension	128	—
Dropout	0.0	0.1
Image size	256	256
Patch size	16	16
Refinement blocks per branch	2	2
Input Preprocessing		
Pixels	$[-1, 1]$ linear rescaling	
Semantic feature	Normalized features scaled by $\beta = 0.5$	
Training		
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$	
Batch size	256	
Learning rate	1×10^{-4}	
Learning rate schedule	constant	
Weight decay	0	
EMA decay	0.9999	
Time sampler	$\mathrm{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = -0.8$, $\sigma = 0.8$	
Semantic loss weight	0.1	
REPA weight	0.5	
REPA feature layer	8	
Noise scale	1.0	
Minimum $(1 - t)$	0.05	
Sampling		
ODE solver	Euler	Heun
ODE steps	50	50
Base time grid	uniform in $[0, 1]$ before timeshift	
Timeshift	1.0	2.0
CFG scale	1.0	2.25
CFG interval	—	$[0.1, 0.9]$

target images. The feature networks remain frozen, with perceptual gradients propagated through the predicted image. The full objective combines the pixel velocity loss, semantic loss, REPA alignment, and perceptual supervision; the semantic and REPA loss weights are 0.1 and 0.5, respectively.

A.4 Semantic Token Construction and SNR Calibration

We extract a 16×16 grid of patch features $F = E(x)$ from a frozen DINOv2-with-registers encoder. The aggregation operator $\mathcal{A}$ averages local 2×4 neighborhoods to form 32 semantic tokens. We apply the same operator to the stored training-set mean and variance arrays of the dense features:

$$\bar{s} = \mathcal{A}(F), \qquad \bar{\mu} = \mathcal{A}(\mu_E), \qquad \bar{v} = \mathcal{A}(v_E). \tag{8}$$

The semantic target is then

$$\tilde{s} = \frac{\bar{s} - \bar{\mu}}{\sqrt{\bar{v} + \varepsilon}}, \qquad s = \beta \tilde{s}, \qquad \varepsilon = 10^{-5}, \quad \beta = 0.5. \tag{9}$$

All operations are applied elementwise at each token position and feature channel. Here, $\bar{v}$ is the spatially aggregated dense-feature variance statistic used for preprocessing. This preprocessing sets the numerical scale of the semantic targets.

Under the joint pixel-semantic noising process,

$$x_{t_x} = t_x x + (1 - t_x)\epsilon_x, \qquad s_{t_s} = t_s s + (1 - t_s)\epsilon_s, \tag{10}$$

where $\epsilon_x, \epsilon_s \sim \mathcal{N}(0, I)$, the effective semantic signal-to-noise ratio (SNR) at time t_s is

$$\mathrm{SNR}_s(t_s) = \frac{\mathbb{E}\left[\|t_s s\|_2^2\right]}{\mathbb{E}\left[\|(1-t_s)\epsilon_s\|_2^2\right]} = \frac{t_s^2 \beta^2 \mathbb{E}\left[\|\tilde{s}\|_2^2\right]}{(1-t_s)^2 \mathbb{E}\left[\|\epsilon_s\|_2^2\right]}. \tag{11}$$

Thus, semantic feature rescaling directly changes the semantic SNR along the denoising trajectory.

The same SNR adjustment can be interpreted equivalently as a timestep shift. Suppose we keep the unscaled semantic tokens $\tilde{s}$ but change their diffusion time from t_s to t'_s, such that the SNR of the unscaled stream at t'_s matches the SNR of the rescaled stream at t_s:

$$\mathrm{SNR}_{\tilde{s}}(t'_s) = \mathrm{SNR}_s(t_s). \tag{12}$$

Using Eq. (11), this gives

$$\frac{t'_s}{1-t'_s} = \beta \frac{t_s}{1-t_s}. \tag{13}$$

Solving for t'_s, we obtain

$$t'_s = \frac{\beta t_s}{1 + (\beta - 1) t_s}. \tag{14}$$

Therefore, rescaling the semantic tokens by β is SNR-equivalent to applying a shifted diffusion schedule to the unscaled semantic stream.

The default joint trajectory applies this signal scaling at a common time for pixels and semantics; Appendix B.9 separately tests schedules that advance the semantic state.

A.5 Sampling with Classifier-free Guidance

Sampling maintains the pixel and semantic states, (x_k, s_k), which are advanced by the ODE solver. The L-sized model evaluations use Euler steps and the XL evaluations use Heun steps. At step k, the clean-target predictions $\hat{x}_k$ and $\hat{s}_k$ are converted into velocities:

$$\hat{v}_{x,k} = \frac{\hat{x}_k - x_k}{1 - t_k}, \qquad \hat{v}_{s,k} = \frac{\hat{s}_k - s_k}{1 - t_k}. \tag{15}$$

At each solver evaluation, the conditional and class-unconditional branches receive the same current pixel and semantic states (x_k, s_k) and differ only in the class condition. The class-unconditional branch replaces the class condition with the null condition used during classifier-free training. CFG is applied to both the pixel and semantic velocities using the same guidance scale and interval.

For all guided XL results, classifier-free guidance uses a scale of 2.25 and an active interval of $[0.1, 0.9]$ for both pixel and semantic velocities. Outside this interval, the guidance scale is 1.0. Unguided evaluation uses a scale of 1.0 throughout the trajectory. XL sampling uses 50 Heun steps with a timeshift of 2.0.

A.6 Implementation Details for Evolving Semantic States

All state-intervention variants in Table 4 are trained and evaluated with the corresponding intervention enabled throughout, rather than modifying a fully trained SemPix model only at inference.

No semantic carryover. The semantic input is reset to zero at every denoiser evaluation during both training and sampling. The semantic prediction target, loss, architecture, and within-step pixel–semantic interaction are unchanged; only the generated semantic state passed from the previous denoising step is removed. This directly tests the effect of propagating the semantic state across denoising evaluations.

Blocked semantic-to-pixel attention. Pixel queries are prevented from attending to semantic keys and values at every layer, while the semantic trajectory continues to evolve normally. All other attention connections are unchanged, and the conditioning tokens do not provide an indirect semantic-to-pixel information path.

Downsampled pixel auxiliary state. We average-pool the RGB image with a 2×4 kernel and stride and partition the result into non-overlapping 16×16 patches, yielding 32 flattened 768-dimensional RGB tokens without an additional projection. The auxiliary state follows the same noising schedule, state-update rule, and token interaction mechanism as the semantic branch, with REPA and perceptual supervision retained.

A.7 Backbone Architecture Details

Both dual-stream variants use 24 layers per stream. We match the parameter budget by reducing the hidden dimension while preserving the depth of cross-stream interaction. The parameter-matched variant uses a hidden dimension of 728 with 14 attention heads, while the full-width variant uses 1024 with 16 heads. Each stream has independent QKV and output projections, MLPs, and AdaLN parameters. At each layer, the projected queries, keys, and values are concatenated along the token dimension for joint self-attention, and the outputs are then separated into their respective streams.

B Additional Ablations

Pixel-only controls. Table 10A evaluates pixel-only controls with a refinement head or without in-context class tokens. Adding a refinement head to PixelGen yields 7.62 FID, compared with 7.53 for the baseline. Refinement alone does not reproduce the improvement to 5.33 achieved by SemPix. Removing the in-context class tokens from PixelGen yields 7.65 FID. These class-conditioning tokens are distinct from the generated semantic state used by SemPix.

Refinement depth. Table 10B compares refinement depths. Two blocks per head achieve the best FID and IS among the tested depths. We use this configuration for both branches.

Table 10: **Additional architectural ablations.** Group A studies pixel-only PixelGen controls; group B varies the number of refinement blocks per branch in SemPix-L. All models are trained for 200K steps.

Configuration	FID ↓	IS ↑	Prec. ↑	Rec. ↑
A. Pixel-only controls				
PixelGen	7.53	131.7	0.72	0.60
PixelGen with refinement	7.62	128.4	0.72	0.60
PixelGen without in-context class tokens	7.65	124.5	0.71	0.61
B. Refinement blocks per branch				
1 block	5.80	150.4	0.74	0.59
2 blocks	**5.33**	**157.2**	**0.75**	0.59
3 blocks	5.71	149.8	0.73	0.59
4 blocks	6.01	148.3	0.72	**0.60**
5 blocks	6.76	142.7	0.71	**0.60**

B.1 Semantic Granularity under a Dual-Stream Backbone

We further test whether the compact-state result is specific to the shared-backbone design by repeating the 32-versus-256-token comparison under the parameter-matched dual-stream backbone. The architecture and training recipe are held fixed, while the 32-token variant uses locally aggregated semantic features and the 256-token variant uses dense patch features. The compact 32-token state achieves 6.81 FID, compared with 6.98 for the dense 256-token state, together with higher IS and Recall. The same FID, IS, and Recall ordering is observed under the shared backbone, extending the compact-state trend across both backbone designs.

Table 11: **Semantic granularity under a parameter-matched dual-stream backbone.** The 32-token state uses locally aggregated semantic features, while the 256-token state uses dense patch features. The dual-stream architecture and training recipe are otherwise unchanged. All models are trained for 200K steps and evaluated without CFG.

Sem. tokens	Params	FID ↓	IS ↑	Prec. ↑	Rec. ↑
32	463.80M	**6.81**	**141.9**	0.72	**0.59**
256	463.97M	6.98	131.5	**0.73**	0.55

B.2 Representation Construction and Semantic Signal Scale

Representation construction. Table 12A compares global, dense, and aggregated visual features. A single CLS token achieves 7.89 FID, compared with 6.74 for dense patch features and 5.33 for local 2D average pooling. Linear interpolation and random-group pooling achieve 5.77 and 5.78 FID, respectively. Both local pooling geometries are effective: 1×8 and 2×4 neighborhoods achieve 5.41 and 5.33 FID. We adopt 2×4 pooling to retain an explicit two-dimensional spatial organization.

These results show that aggregation itself provides a substantial improvement over dense features, while preserving local spatial organization yields an additional gain among the tested 32-token constructions. We therefore use local 2×4 pooling as the default representation.

Table 12: **Representation construction and matched semantic signal scale.** SemPix-L at 200K steps without CFG.

Representation	Tokens	FID ↓	IS ↑	Prec. ↑	Rec. ↑
A. Representation construction					
Single global token	1	7.89	136.7	0.67	0.58
Dense patch features	256	6.74	131.7	0.74	0.55
Linear interpolation	32	5.77	152.4	0.73	0.60
Local 1D average pooling	32	5.41	158.6	0.74	0.58
Local 2D average pooling	32	5.33	157.2	0.75	0.59
Random-group average pooling	32	5.78	159.1	0.74	0.59
B. Matched semantic signal scale					
Local 2D pooling, rematched scale	32	5.39	157.0	0.74	0.59

Semantic signal scale. Table 12A,B compares dense and locally pooled representations at a common semantic signal scale. The locally pooled 32-token representation achieves 5.39 FID and 0.59 Recall, compared with 6.74 and 0.55 for dense features. The default and post-pooling variance configurations achieve similar FID, 5.33 and 5.39. Experiment details are provided in Appendix A.4 and B.6.

Pixel–semantic consistency. We measure consistency as the mean token-wise cosine similarity between the predicted clean semantic state $\hat{s}_t$ and 32 locally pooled DINOv2 tokens extracted from the corresponding clean-image prediction $\hat{x}_t$. Matched trajectories are compared with independently generated same-class trajectories and a fixed different-class permutation. Curves average 2,000 trajectories from 1,000 ImageNet classes, and shaded regions show pointwise bootstrap 95% confidence intervals.

B.3 Semantic Predictability

Figure 6 compares semantic prediction across noise levels. Since local aggregation changes the variance of the target features, we use variance-normalized MSE together with token-wise cosine similarity. Locally aggregated semantics are more reliably recovered in the high-noise regime, while the difference diminishes as the states become cleaner. The same early advantage is largely preserved after expanding the aggregated representation back to 256 spatial positions, consistent with our matched-length generation results and showing that the effect follows the aggregated representation rather than simply the number of semantic tokens. Together with the semantic-carryover and semantic-to-pixel controls, this supports compact aggregated semantics as an effective context to establish early and carry through pixel generation.

B.4 Semantic Encoder and Token Granularity

Table 13 compares dense and compact semantic states from two visual encoders. Reducing dense 256-token features to 32 locally aggregated tokens improves FID from 6.74 to 5.33 with DINOv2-reg and from 6.37 to 5.19 with SigLIP. Local aggregation also improves IS and Recall for both encoders, supporting the effectiveness of compact semantic context across these two visual representations.

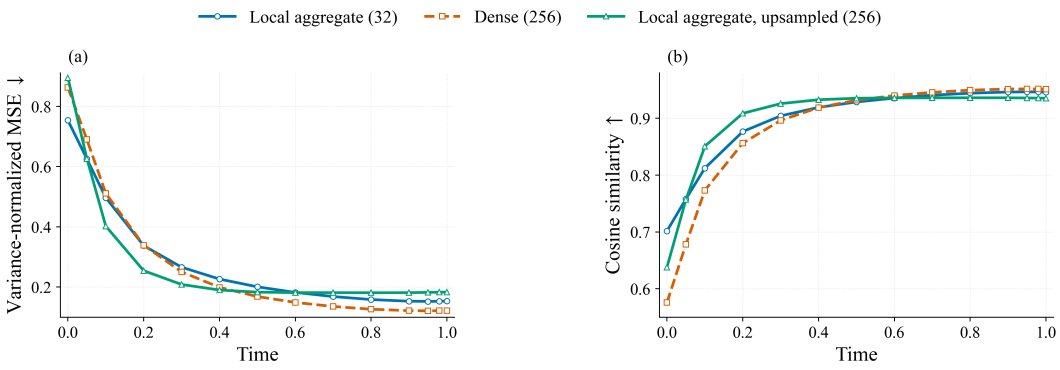


Figure 6: **Semantic predictability along the generation trajectory.** Local aggregation improves semantic recoverability in the high-noise regime, measured by variance-normalized MSE and cosine similarity. Expanding the aggregated representation back to 256 positions largely preserves the early advantage, showing that the effect follows the aggregated representation rather than sequence length alone.

Table 13: **Semantic encoder and token granularity.** Results at 200K steps on ImageNet-256 without CFG. Local aggregation consistently improves over dense semantic features for both DINOv2-reg and SigLIP.

Semantic encoder	Representation	Tokens	FID ↓	IS ↑	Prec. ↑	Rec. ↑
DINOv2-B w/ registers	Dense	256	6.74	131.7	0.74	0.55
DINOv2-B w/ registers	Local aggregation	32	**5.33**	**157.2**	**0.75**	**0.59**
SigLIP-B/16-256	Dense	256	6.37	136.2	0.74	0.56
SigLIP-B/16-256	Local aggregation	32	**5.19**	**155.2**	0.74	**0.58**

B.5 Compact Semantic States over Longer Training

Figure 7 compares FID convergence with 32 and 256 semantic tokens. The 32-token model maintains lower FID as training progresses, indicating that the benefit of semantic compression extends beyond faster early convergence. Between 400K and 600K training steps, its FID remains between 4.29 and 4.55, compared with 5.63–5.72 for the 256-token model. Additional training does not close this gap, showing that compact semantic representations continue to support higher-quality generation over longer training schedules.

B.6 Semantic Normalization

Table 12A,B reports the spatial-pooling comparison between default preprocessing and post-pooling variance normalization. We also evaluate the order of normalization and compression, as well as statistics computed directly from compressed features. Linear interpolation is applied along the token dimension. Normalizing dense features before interpolation yields 6.26 FID and 0.62 Recall. With compression first, the original and recomputed mean-and-variance configurations achieve 5.77 and 5.95 FID. For spatial pooling, the default preprocessing and the post-pooling variance configuration achieve similar FID, 5.33 and 5.39. Compact semantic representations thus remain effective under multiple statistical preprocessing choices.

B.7 Semantic Prediction Objective

Table 14 compares semantic prediction targets and temporal weighting on SemPix-L with 32 semantic tokens at 200K steps. Temporal weighting substantially improves clean semantic prediction, reducing FID to 5.33. The adopted $(1-t)^{-1}$ weighting also outperforms direct velocity prediction and the $(1-t)^{-2}$-weighted clean-target objective in this comparison. Compared with the latter, it improves IS from 147.8 to 157.2 and Precision from 0.73 to 0.75.

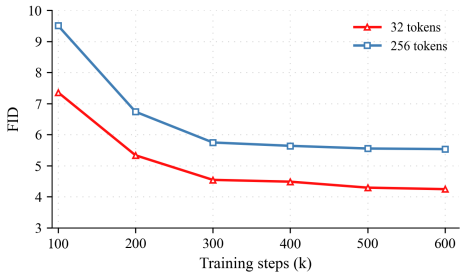


Figure 7: **FID convergence with different numbers of semantic tokens.** The compact 32-token semantic representation consistently achieves lower FID than the 256-token representation from 100K to 600K training steps.

Table 14: Ablation of semantic denoising design on the L-sized model at 200K steps.

Prediction	loss	FID ↓	IS ↑	Prec. ↑	Rec. ↑
v-prediction	v-loss	5.81	154.1	0.73	0.58
x-prediction	v-loss	5.49	147.8	0.73	0.58
x-prediction	$(1-t)$ v-loss	**5.33**	**157.2**	**0.75**	0.59

The velocity loss corresponds to the $(1-t)^{-2}$-weighted x-loss velocity objective. Appendix B.8 and B.9 report additional ablations of the semantic loss weight and semantic-leading schedules.

B.8 Semantic Loss Weight Sensitivity

Table 15 studies the sensitivity to the semantic loss weight in SemPix-L with 32 spatially pooled semantic tokens and dual refinement heads. Across weights from 0.05 to 0.50, FID ranges from 5.33 to 6.07, indicating that the model is not highly sensitive to this hyperparameter. The default weight $\lambda_{\text{sem}} = 0.1$ achieves the best FID and IS among the tested settings, and we use it throughout our experiments.

Table 15: **Semantic loss weight sensitivity.** SemPix-L with 32 spatially pooled semantic tokens and dual refinement heads, trained for 200K steps and evaluated without CFG.

λ_{sem}	FID ↓	IS ↑	Prec. ↑	Rec. ↑
0.05	5.39	149.3	0.73	0.58
0.10	**5.33**	**157.2**	**0.75**	**0.59**
0.20	5.53	147.1	0.74	**0.59**
0.30	5.66	143.5	0.72	0.58
0.40	5.85	143.1	0.72	0.58
0.50	6.07	138.5	0.70	0.57

B.9 Semantic Update Schedule

We also ask whether the semantic state needs to advance ahead of the pixel state along the generation trajectory. Table 16 compares synchronous updates with semantic-leading schedules at different temporal offsets δ. Synchronous updates achieve the best FID and substantially higher Recall, while semantic-leading schedules trade coverage for higher IS or Precision. We therefore use synchronous updates by default. This result indicates that SemPix does not require an explicitly semantic-first trajectory for the generated semantic state to benefit pixel generation.

B.10 Higher-resolution and Text-to-Image Generation

Higher-resolution generation. Table 17 compares class-conditional ImageNet-512 generation without classifier-free guidance. At the same 400K-step (80-epoch) training budget, SemPix-L

Table 16: **Semantic update schedule.** We compare synchronous pixel–semantic updates with schedules that advance the semantic state ahead of the pixel state. All models use SemPix-L with 32 semantic tokens and are trained for 200K steps without CFG.

Schedule	FID ↓	IS ↑	Prec. ↑	Rec. ↑
Synchronous	**5.33**	157.2	0.75	**0.59**
Semantic-leading, $\delta = 0.10$	5.72	**165.5**	0.76	0.51
Semantic-leading, $\delta = 0.20$	5.67	163.2	0.76	0.50
Semantic-leading, $\delta = 0.30$	5.84	161.6	**0.78**	0.45

Table 17: ImageNet-512 class-conditional generation without CFG at 400K steps (80 epochs). SemPix-L uses 32 semantic tokens with 2D pooling.

Method	FID ↓	IS ↑	Prec. ↑	Rec. ↑
JiT-L	29.04	55.5	0.50	0.55
PixelGen-L	7.31	152.5	0.78	**0.59**
SemPix-L	**5.61**	**167.0**	**0.79**	**0.59**

achieves an FID of 5.61, compared with 7.31 for PixelGen-L and 29.04 for JiT-L. In particular, SemPix reduces FID by 23.2% relative to PixelGen. Together with the ImageNet-256 results, this experiment shows that the benefit of evolving semantic context persists at a higher image resolution.

Text-to-image generation. We extend SemPix to 256×256 text-to-image generation on MS-COCO 2014, training from scratch on train2014 with frozen CLIP ViT-L/14 text conditioning. The baseline uses 12 backbone blocks and 32 text in-context tokens; SemPix uses 10 shared blocks, one refinement block per head, and 32 generated semantic tokens. Semantic targets are locally pooled DINOv2 features. Both models train for 200K steps with batch size 256, without REPA or perceptual losses. We evaluate EMA checkpoints with 50-step Euler sampling and CFG 2.0, using the same 50K val2014 captions and pixel-noise seeds; FID uses all 40,504 val2014 images as its reference.

Table 18: **Text-to-image generation on MS-COCO 2014.** CLIP cosine is the raw cosine similarity; PickScore is the mean scaled cosine score.

Method	Params	FID ↓	CLIP cosine ↑	PickScore ↑
Baseline	252.02M	30.47	0.21281	18.9523
SemPix	233.91M	**23.48**	**0.21818**	**19.1202**

Table 18 compares SemPix with the pixel-only baseline on MS-COCO text-to-image generation. SemPix reduces FID by 22.9% and improves both text-image alignment and PickScore, with fewer trainable parameters. These results support extending generated semantic context to text-conditioned pixel generation.

C Computational Cost

Table 19 compares practical generation cost under a unified measurement setup. All models are measured on the same hardware and software stack using the same timing protocol. With 32 semantic tokens, SemPix-L matches PixelGen in training time, memory usage, and per-image inference time while improving FID from 7.53 to 5.33. In contrast, using 256 semantic tokens substantially increases computation and memory while also degrading generation quality, highlighting the practical advantage of compact semantic context.

D From Latent Representations to Pixel Generation

Latent-space diffusion and flow models learn to generate in an autoencoder's representation space and decode the resulting latents into images Rombach et al. (2022); Peebles & Xie (2023); Ma et al.

Table 19: **Computational cost of compact semantic context.** L-sized models on ImageNet-256 under the 200K-step controlled setting. All timing and memory measurements are obtained under the same hardware, software environment, and measurement protocol. The SemPix variants with 32 and 256 semantic tokens compare the computational cost of compact and dense semantic states. All models use 50-step Euler sampling without CFG.

Method	Sem. tokens	Train (s/step) ↓	Memory (GB) ↓	Inference (s/image) ↓	FID ↓
JiT	–	0.19	29.5	0.32	23.64
PixelGen	–	0.20	33.9	0.32	7.53
SemPix-L	32	0.20	33.9	0.32	**5.33**
SemPix-L	256	0.33	45.0	0.47	6.74

(2024); Esser et al. (2024). Building on Transformer architectures Vaswani et al. (2017), this direction has improved efficiency through masked training Zheng et al. (2024); Gao et al. (2023), training strategies Yao et al. (2024), and alternative backbones Wang et al. (2024b; 2026b). The representation space also affects generation quality, motivating autoencoders with improved diffusability Skorokhodov et al. (2025) and latent representations informed by pretrained visual encoders Yao et al. (2025); Zheng et al. (2026). Visual and textual representations further support multimodal understanding Wang et al. (2024a) and unified or instruction-conditioned generation Xie et al. (2025); Zhou et al. (2025); Wu et al. (2026); Lin et al. (2025).

Pixel-space generators instead synthesize images directly without an image autoencoder, from early diffusion and score-based models and text-to-image cascades Ho et al. (2020); Song & Ermon (2019); Song et al. (2021); Dhariwal & Nichol (2021); Saharia et al. (2022) to recent scalable pixel-space architectures Hoogeboom et al. (2023); Chen et al. (2025); Wang et al. (2026a); Li & He (2026); Yu et al. (2026); Chen et al. (2026); Ma et al. (2026a). Pretrained visual features can still support direct pixel generation through representation alignment Yu et al. (2025) and perceptual supervision Zhang et al. (2018); Ma et al. (2026b), motivating the study of how visual semantics can participate in the pixel-generation process.

E Positioning among Generative Visual-State Methods

Recent methods establish that pretrained visual representations can be made explicit variables in the generative process. ReDi Kouzelis et al. (2025) jointly synthesizes image latents and dense visual features, REG Wu et al. (2025) couples latent generation with a global semantic token, and V-Co Lin et al. (2026) studies dense visual co-denoising directly in pixel space. SemPix builds on this broader direction but asks a different question: *what semantic state should be carried in direct pixel generation, and what properties make that state useful as context for subsequent pixel prediction?*

Our study separates this question into two axes. We first study the *form* of the carried state, comparing global, coarse spatial, and dense semantic representations while controlling for sequence length and integration mechanism. We then study its *trajectory role*, directly intervening on cross-step persistence and semantic-to-pixel access. SemPix advances this direction by making the carried semantic state itself the object of study. We systematically examine its representation form and trajectory role in direct pixel generation, using matched-length, integration, carryover, and interaction controls.

Table 20: **Positioning among generative visual-state methods.** Prior work establishes joint generation of image and visual states, while SemPix focuses on the form and trajectory role of the carried semantic state.

Method	Image state	Generated visual state	Central question
ReDi Kouzelis et al. (2025)	VAE latent	Dense visual features	Joint image–feature synthesis
REG Wu et al. (2025)	VAE latent	Global semantic token	Semantic entanglement
V-Co Lin et al. (2026)	RGB pixels	Dense visual features	Visual co-denoising design
SemPix	**RGB pixels**	**Coarse spatial semantics**	**State form and trajectory context**

The distinction in Table 20 is therefore primarily one of research question rather than whether visual representations are generated at all. SemPix systematically varies the abstraction level of the semantic state and uses matched-length controls to separate representation content from sequence length, while the no-carryover and blocked semantic-to-pixel variants test the effects of removing cross-step propagation and semantic-to-pixel access. These experiments make the design of the carried semantic state itself the object of study.

F SAMPLING TRAJECTORIES

In Figures 8 and 9, we visualize the sampling trajectories of JiT, JiT with REPA and perceptual losses, and SemPix on ImageNet examples. For each example, all methods share the same class condition and initial pixel noise. At intermediate timesteps, we display the predicted clean image, allowing object structure and appearance to be inspected without the obscuring effect of the noisy sampling state. These examples provide a qualitative comparison of how the methods form and refine image content.



Figure 8: Sampling trajectories comparison.



Figure 9: Sampling trajectories comparison.

G More Qualitative Results

In Figures 10 and 11, we present ImageNet-256 samples generated by SemPix-XL/16 after 160 epochs of training, conditioned on the specified classes.

H Discussion

SemPix points to generated context as an additional design axis for iterative image generation. Existing generators are typically organized around what is predicted and how external conditions are injected. Our results suggest that what the model explicitly carries between generation steps is another important choice. A compact semantic state can evolve with the emerging image and complement both pixel states and external conditioning. The higher-resolution and text-to-image results suggest that this principle extends beyond class-conditional ImageNet generation. Looking forward, it will be interesting to study how generated context should scale in capacity and spatial granularity, and how multiple forms of internal and external context can interact in larger-scale, high-resolution multimodal generators.

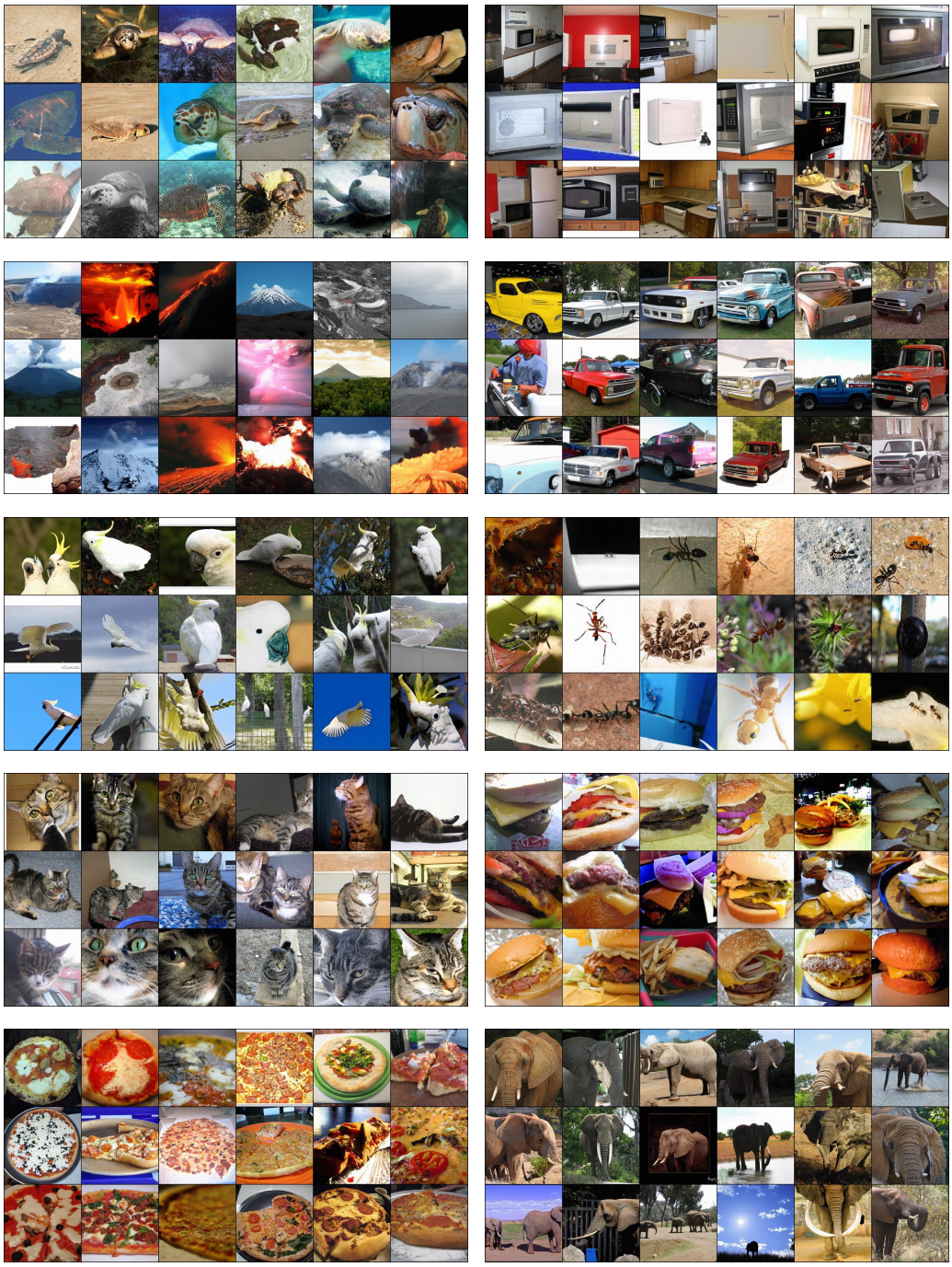


Figure 10: **SemPix-XL samples on ImageNet-256.** Class-conditional images generated after 160 training epochs with classifier-free guidance.

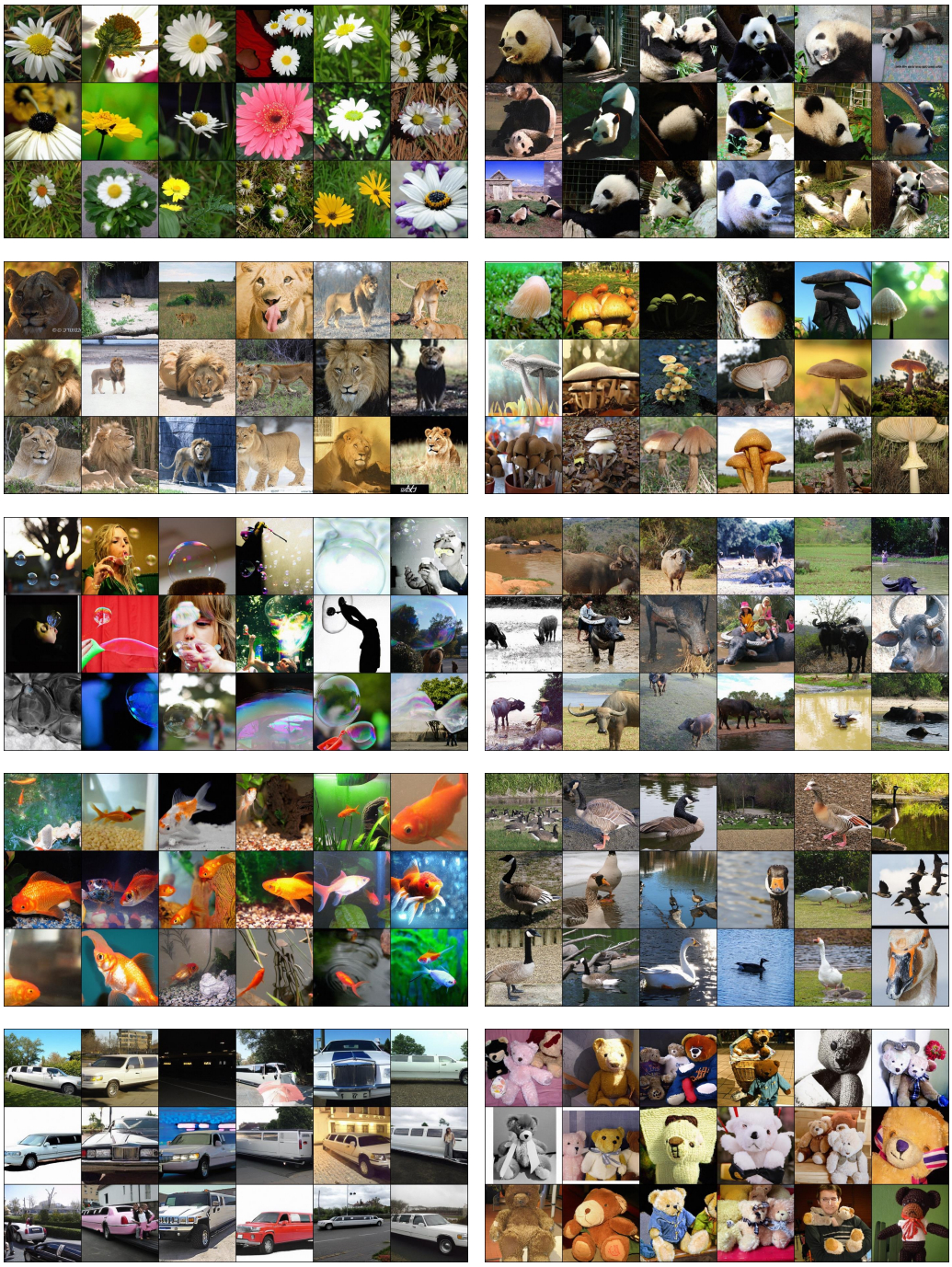


Figure 11: **SemPix-XL samples on ImageNet-256.** Class-conditional images generated after 160 training epochs with classifier-free guidance.