PATCHBRIDGE: DECOUPLING PATCH ROLES FOR PIXEL-SPACE DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel-space diffusion Transformers denoise RGB images directly, avoiding the fixed autoencoder compression interface common in latent diffusion. Yet many process one patch grid throughout the backbone, asking it to support global context modeling, full-depth denoising, and late local refinement. These roles differ in their spatial and computational requirements; we call their forced sharing of one grid patch-role coupling. We introduce PatchBridge, which assigns role-specific patch sizes and lifetimes to three coordinated streams. A compact semantic stream maintains global context through sparse updates, a structural stream carries the principal denoising state through the full backbone, and a finer detail stream activates in later layers for local refinement. This allocation preserves full-depth denoising without processing the finest grid at every layer. Parent-to-child spatial modulation supplies aligned context to finer streams, while child-to-parent pooled feedback returns updated evidence to coarser streams, keeping their representations coordinated as views of one evolving denoising state. At the output, an overlapping clean-image readout decouples detail-token stride from prediction support, allowing neighboring tokens to jointly estimate shared pixels. PatchBridge achieves FIDs of 1.44 and 1.54 on class-conditional ImageNet at 256×256 and 512×512, respectively. In text-to-image generation, it scores 0.87 on GenEval and 84.1 on DPG-Bench. These results support role-specific patch allocation for direct pixel-space generation.

1 INTRODUCTION

Pixel-space diffusion denoises images directly, avoiding the fixed autoencoder compression interface used by latent-space diffusion models (Rombach et al., 2022; Peebles & Xie, 2023). JiT (Li & He, 2026) shows that a plain Transformer can generate high-quality images from large pixel patches. Large patches keep the token sequence manageable, but a single grid forces global context modeling, full-depth denoising, and local detail refinement to use the same spatial granularity. These jobs place different demands on spatial resolution and processing depth. We ask whether a pixel-space denoiser should make all three share one patch grid.

Large patches reduce token count and make global reasoning efficient, but provide coarser spatial granularity. Small patches preserve finer spatial structure, but make full-depth processing expensive. These roles also differ in depth: global context can be updated intermittently, the structural denoising state must persist throughout the backbone, and fine detail is most useful near the output. A single grid forces all three roles to share one patch size throughout the network, tying the resolution available for local refinement to the computational cost of full-depth denoising. We call this constraint patch-role coupling.

Prior pixel-space methods address computational cost through noise-schedule design, multi-resolution processing, architectural downsampling, and cascaded generation (Hoogeboom et al., 2023; Chen et al., 2025c). Recent architectures also use specialized pixel decoders or frequency-aware processing to balance global structure and local fidelity (Yu et al., 2026; Ma et al., 2026a). In contrast, we organize persistent representations at different patch scales within a single denoising backbone, assigning them role-specific processing depths and update frequencies. Reciprocal interactions allow finer-scale evidence to update coarser states before subsequent computation, keeping these representations coordinated throughout their active lifetimes.

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

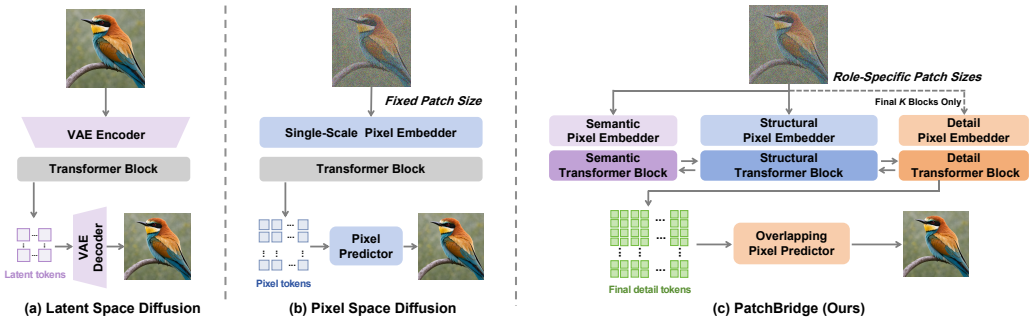


Figure 1: Comparison of latent-space diffusion, single-scale pixel-space diffusion, and PatchBridge. PatchBridge uses role-specific patch sizes for semantic, structural, and detail processing, with late-stage detail refinement and overlapping clean-image readout.

Building on this insight, we propose PatchBridge, which assigns different patch scales and lifetimes to three streams. The structural stream carries the main denoising state through the full Transformer backbone at an intermediate patch size. A compact semantic stream maintains global context on a coarser grid and is refreshed at sparse macro stages. A finer detail stream activates in later stages to refine local shapes, boundaries, and textures using the structural context already accumulated. This allocation retains full-depth denoising without processing the finest grid in every layer. Fig. 1 contrasts PatchBridge with a single-scale pixel Transformer.

Distributing these roles across distinct streams requires active coordination. Coarse global context must guide finer computations, and newly refined local details must update the persistent coarser states. To achieve this, PatchBridge connects adjacent streams through reciprocal cross-scale interactions. Downward spatial modulation provides the finer child stream with aligned global context, while upward feedback aggregates the refined local features to update the coarser parent stream. Unlike traditional coarse-to-fine cascades that operate sequentially, this bidirectional communication ensures all streams evolve synchronously, jointly updating a single, unified denoising state.

PatchBridge derives its clean-image prediction from the detail stream. Standard non-overlapping readouts force each token to independently predict a disjoint patch, which causes visible grid artifacts at patch boundaries. To address this, PatchBridge introduces an overlapping readout: each detail token predicts a region wider than its spatial stride. These overlapping predictions are combined through coverage-normalized folding. By allowing neighboring tokens to jointly determine shared boundary pixels, this strategy smooths transitions and produces a cohesive final image.

We test whether the proposed components contribute under matched training settings. In the 800-epoch B-model ablation, adding the semantic stream, late detail stream, and overlapping readout reduces FID from 3.68 to 2.91. Separate feedback and overlap ablations test the two coordination mechanisms. At larger scale, PatchBridge reaches 1.44 FID on ImageNet 256×256 and 1.54 FID at 512×512. The text-to-image model reaches 0.87 on GenEval and 84.1 on DPG-Bench. Together, these results evaluate role specialization across model sizes, resolutions, and generation settings.

Our contributions are summarized as follows:

- We introduce PatchBridge, a pixel-space diffusion Transformer that assigns global context modeling, full-depth denoising, and late detail refinement to semantic, structural, and detail streams with role-specific patch sizes and lifetimes. This allocation preserves a persistent denoising backbone without processing the finest token grid at every layer.
- We coordinate adjacent streams through reciprocal cross-scale interactions, allowing coarse context to modulate finer computation and pooled fine-scale features to update coarser states. An overlapping clean-image readout lets neighboring detail tokens jointly predict shared pixels, extending coordination from internal features to the final output.
- Among the compared pixel-space models, PatchBridge achieves the best FID and Inception Score on class-conditional ImageNet at both 256×256 and 512×512, as well as the highest GenEval and DPG-Bench scores in text-to-image generation. These results demonstrate gains across resolutions and conditioning settings.

2 Related Works

Diffusion models Ho et al. (2020); Song et al. (2021) generate images by iteratively denoising noise. Their denoisers can operate in an autoencoder-compressed latent space or directly in RGB space, a choice that affects both computation and the image information available to the model.

2.1 Latent Diffusion Models

Latent diffusion models (LDMs) Rombach et al. (2022); Podell et al. (2024) reduce cost by denoising autoencoder-compressed representations Kingma & Welling (2014); van den Oord et al. (2017). DiT Peebles & Xie (2023) and SiT Ma et al. (2024) establish scalable Transformer backbones in this space. REPA Yu et al. (2025) aligns diffusion features with pretrained visual encoders, and REPA-E Leng et al. (2025) extends alignment to end-to-end optimization. VA-VAE Yao et al. (2025) and RAE Zheng et al. (2026) build richer latent spaces from foundation-model representations. These methods still depend on an intermediate learned representation.

2.2 Pixel-Space Diffusion Models

Early pixel-space diffusion models use U-Net backbones Ho et al. (2020); Dhariwal & Nichol (2021), whose cost grows with image resolution. Multi-resolution and hierarchical approaches improve scalability Hoogeboom et al. (2025); Chen et al. (2025c). JiT Li & He (2026) achieves strong pixel-space generation with large patches and clean-image prediction. PixelDiT Yu et al. (2026) pairs patch-level semantics with pixel-level refinement; DeCo Ma et al. (2026a) separates low-frequency semantics from high-frequency synthesis; and PixelU Guo et al. (2026) uses hierarchical Transformer processing for global modeling and local refinement. These methods balance global efficiency and local detail through different designs.

3 Method

PatchBridge assigns global context modeling, full-depth denoising, and local detail refinement to the semantic, structural, and detail streams, respectively. We first define the denoising task and the asymmetric lifecycles of these representations, then describe their reciprocal interactions and overlapping image readout.

3.1 Denoising Formulation and Overview

Let $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ be a clean image and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ be Gaussian noise, where σ is the noise scale. Following JiT (Li & He, 2026), we interpolate from noise to data and directly predict the clean image:

$$\mathbf{x}_t = t\mathbf{x} + (1-t)\boldsymbol{\epsilon}, \qquad \widehat{\mathbf{x}} = F_\theta(\mathbf{x}_t, t, y), \quad t \in [0,1]. \tag{1}$$

Here y is the generation condition. We use $\mathbf{x}$ for the clean image to distinguish it from the noisy state at $t = 0$. All three streams operate within a single evaluation of F_θ at a fixed diffusion timestep; their lifecycles refer to network depth, not to different portions of the sampling trajectory.

The semantic, structural, and detail streams use patch sizes $p_{\mathrm{sem}} > p_{\mathrm{str}} > p_{\mathrm{det}}$, respectively. For $a \in \{\mathrm{sem}, \mathrm{str}, \mathrm{det}\}$, let $\mathbf{Z}^a \in \mathbb{R}^{N_a\times d_a}$ denote the image-token state, where $N_a = HW/p_a^2$ and d_a is the feature dimension of each token. The patch sizes divide H and W; adjacent patch-size ratios are integers, set to two in our models. We consistently write states as token sequences: Grid_a reshapes a sequence into its spatial grid, and Vec reverses this operation.

The structural stream retains all L JiT blocks and carries the principal denoising state. The semantic stream uses M global-attention blocks distributed across this depth, reusing compact context between updates. The detail stream pairs a local-attention block with each of the final K structural blocks, allocating dense-token computation to late refinement. All streams are jointly trained through a single clean-image prediction objective. For class conditioning, $\mathbf{c} = \mathcal{T}(t) + \mathcal{Y}(y)$ combines an MLP-transformed sinusoidal timestep embedding with a learned class embedding. Separate learned linear projections map $\mathbf{c}$ to the semantic and detail feature dimensions. For text conditioning, MiniT2I replaces the class embedding with a linearly projected mean-pooled text representation and

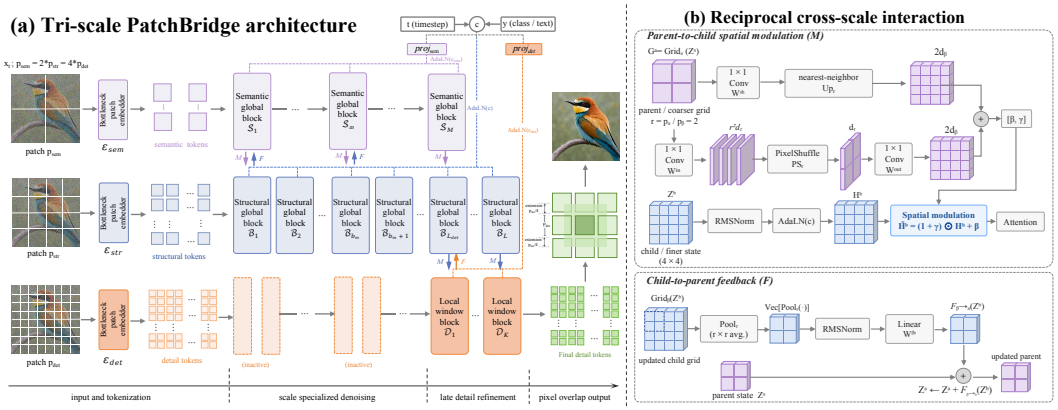


Figure 2: Overview of PatchBridge. (a) Streams coordinate global context modeling, full-depth denoising, and late-stage refinement before overlapping clean-image readout. (b) Adjacent streams communicate via parent-to-child spatial modulation and child-to-parent feedback.

uses projected text tokens for in-context conditioning. We retain each backbone's in-context conditioning; cross-scale operations act only on image tokens.

3.2 Asymmetric Tri-Scale Representations

All three streams obtain their initial embeddings from the same noisy image:

$$\mathbf{Z}_0^a = \mathcal{E}_a(\mathbf{x}_t) + \mathbf{P}_a, \qquad a \in \{\mathrm{sem}, \mathrm{str}, \mathrm{det}\}, \tag{2}$$

where $\mathcal{E}_a$ is a bottleneck patch embedder and $\mathbf{P}_a$ is a fixed two-dimensional positional embedding. The semantic and structural streams start from these embeddings at the beginning of the backbone. The detail stream remains inactive until the final K structural blocks, when its initial embedding is combined with the current structural features. Let $1 = b_1 < \cdots < b_M \le L$ denote the semantic update boundaries, set $b_{M+1} = L+1$, and define the structural intervals as $\mathcal{I}_m = \{b_m, \ldots, b_{m+1} - 1\}$. The semantic state persists across these intervals, providing reusable context without a second full-depth backbone.

The detail stream becomes active immediately after structural block $\ell_{\mathrm{det}} = L - K + 1$. At activation, its initial image embedding is augmented with the current structural features:

$$\mathbf{Z}^{\mathrm{det}} = \mathbf{Z}_0^{\mathrm{det}} + \mathrm{Vec}\left[\mathrm{PS}_{r_{\mathrm{det}}}\left(\mathrm{Grid}_{\mathrm{str}}(\mathbf{Z}^{\mathrm{str}})\mathbf{W}^{\mathrm{init}}\right)\right], \tag{3}$$

where $r_{\mathrm{det}} = p_{\mathrm{str}}/p_{\mathrm{det}}$ and $\mathbf{W}^{\mathrm{init}} \in \mathbb{R}^{d_{\mathrm{str}} \times r_{\mathrm{det}}^2 d_{\mathrm{det}}}$. The projection produces $r_{\mathrm{det}}^2 d_{\mathrm{det}}$ channels at each structural grid location. Pixel shuffle $\mathrm{PS}_{r_{\mathrm{det}}}$ rearranges these channels into an $r_{\mathrm{det}} \times r_{\mathrm{det}}$ group of spatially aligned detail tokens, each with d_{det} features. Local refinement thus starts with both fine-resolution image evidence and accumulated structural context. Detail blocks use self-attention within fixed, non-overlapping $w \times w$ token windows, and only their final state is decoded into an image.

3.3 Reciprocal Cross-Scale Coordination

Each adjacent pair uses complementary operations: the coarser parent modulates the finer child's computation, and the updated child returns spatially pooled features to the parent. This applies to semantic–structural and structural–detail pairs, with no direct semantic–detail connection.

Parent-to-child modulation. For parent a and child b with patch-size ratio $r = p_a/p_b$, define $\mathbf{G}^a = \mathrm{Grid}_a(\mathbf{Z}^a)$. A shared-plus-delta mapping produces spatial shifts and scales:

$$(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \mathcal{M}_{a\to b}(\mathbf{Z}^a) = \mathrm{Split}\left(\mathrm{Vec}\left[\mathrm{Up}_r(\mathbf{G}^a\mathbf{W}^{\mathrm{sh}}) + \mathrm{PS}_r(\mathbf{G}^a\mathbf{W}^{\mathrm{in}})\mathbf{W}^{\mathrm{out}}\right]\right). \tag{4}$$

The shared projection $\mathbf{W}^{\mathrm{sh}} \in \mathbb{R}^{d_a \times 2d_b}$ produces a common modulation for each parent's $r \times r$ children, replicated by nearest-neighbor upsampling Up_r. The delta path projects parent features

through $\mathbf{W}^{\text{in}} \in \mathbb{R}^{d_a \times r^2 d_r}$, uses pixel shuffle to distribute them to child locations, and applies $\mathbf{W}^{\text{out}} \in \mathbb{R}^{d_r \times 2d_b}$ to produce child-specific corrections, where d_r is the bottleneck feature dimension. The two paths are summed on the child grid and flattened by Vec. Split then divides the $2d_b$ feature channels into equal parts, yielding $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^{N_b \times d_b}$. Each semantic stage and detail block has its own mapping parameters; stage indices and biases are omitted for clarity.

Let $\mathbf{H}^b$ denote the child attention input after RMS normalization and timestep/condition AdaLN modulation. We apply the spatial modulation as

$$\widetilde{\mathbf{H}}^b = (1 + \boldsymbol{\gamma}) \odot \mathbf{H}^b + \boldsymbol{\beta}. \tag{5}$$

The attention residual gate and condition-modulated MLP retain the standard backbone update. Detail blocks use non-overlapping $w \times w$ window attention while otherwise following the same block design. Parent information is injected into the child stream through spatial modulation, without token concatenation or cross-attention.

Child-to-parent feedback. The reverse mapping pools aligned child regions, normalizes the resulting features, and projects them to the parent width:

$$\mathcal{F}_{b \to a}(\mathbf{Z}^b) = \text{RMSNorm}\Big(\text{Vec}\big[\text{Pool}_r(\text{Grid}_b(\mathbf{Z}^b))\big]\Big)\mathbf{W}^{\text{fb}}, \qquad \mathbf{W}^{\text{fb}} \in \mathbb{R}^{d_b \times d_a}. \tag{6}$$

Here Pool_r averages each spatially aligned $r \times r$ child group. The resulting residual updates the parent's image-token state with newly computed child information. Unlike downward modulation, which changes how the child processes its features, this reverse path allows adjacent representations to evolve reciprocally rather than through a one-way coarse-to-fine cascade. Both parent-to-child modulation and child-to-parent feedback are illustrated in Fig. 2 (b).

3.4 Overlapping Clean-Image Readout

The readout decouples detail-token stride p_{det} from prediction support. A condition-modulated normalization and linear head map each final detail token i to a $q \times q$ RGB field $\widehat{\mathbf{R}}_i$, with $q > p_{\text{det}}$. Fields are placed at stride p_{det} and combined by coverage-normalized folding:

$$\begin{aligned} \widehat{\mathbf{x}}(\mathbf{u}) &= c(\mathbf{u})^{-1} \sum_{i:\, \mathbf{u} \in \Omega_i} \widehat{\mathbf{R}}_i(\mathbf{u}), \\ c(\mathbf{u}) &= \sum_i \mathbf{1}\{\mathbf{u} \in \Omega_i\}. \end{aligned} \tag{7}$$

Here Ω_i is the support of field i, and $c(\mathbf{u})$ counts how many fields cover image coordinate $\mathbf{u}$. The normalization averages their predictions even where coverage differs near image boundaries. For even $q - p_{\text{det}}$, we fold onto an $(H + q - p_{\text{det}}) \times (W + q - p_{\text{det}})$ canvas and crop $(q - p_{\text{det}})/2$ pixels from each side after normalization. The case $q = p_{\text{det}}$ recovers disjoint patch readout.

Neighboring tokens thereby contribute to shared output pixels rather than owning disjoint regions. We supervise the aggregated clean-image estimate, without a separate loss on individual fields or an explicit agreement constraint. Clean-image parameterization places this output readout directly in RGB space and retains the prediction convention of the underlying JiT model.

Algorithm 1 PatchBridge at timestep t

Require: $\mathbf{x}_t$, y, stage boundaries
$s, h \leftarrow$ initialize by Eq. (2)
for $m = 1, \ldots, M$ **do**
  $s \leftarrow \mathcal{S}_m(s)$
  $\mu_s \leftarrow \mathcal{M}^m_{\text{sem}\to\text{str}}(s)$
  for $\ell = b_m, \ldots, b_{m+1} - 1$ **do**
    $h \leftarrow \mathcal{B}_\ell(h; \mu_s)$
    if $\ell \geq \ell_{\text{det}}$ **then**
      $j \leftarrow \ell - \ell_{\text{det}} + 1$
      if $j = 1$ **then**
        $d \leftarrow$ initialize by Eq. (3)
      end if
      $\mu_d \leftarrow \mathcal{M}^j_{\text{str}\to\text{det}}(h)$
      $d \leftarrow \mathcal{D}_j(d; \mu_d)$
      if $\ell < L$ **then**
        $h \leftarrow h + \mathcal{F}^j_{\text{det}\to\text{str}}(d)$
      end if
    end if
  end for
  if $m < M$ **then**
    $s \leftarrow s + \mathcal{F}^m_{\text{str}\to\text{sem}}(h)$
  end if
end for
$\widehat{\mathbf{x}} \leftarrow \text{Readout}(d; t, y)$
return $\widehat{\mathbf{x}}$

3.5 End-to-End Denoising Procedure

Algorithm 1 specifies the execution order of the components above within one denoiser evaluation at fixed t. We abbreviate the semantic, structural, and detail states by s, h, and d, and use μ_s and μ_d

Table 1: Class-conditional ImageNet 256×256 generation. $\times 2$ denotes classifier-free guidance; for latent models, the second parameter count denotes the pretrained autoencoder.

Method	Epochs	Sampling	Params	FID↓ w/ CFG	IS↑	Precision↑	Recall↑
Latent-space diffusion models							
DiT-XL/2 Peebles & Xie (2023)	1400	250×2	675M+86M	2.27	278.2	0.83	0.57
SiT-XL/2 Ma et al. (2024)	1400	250×2	675M+86M	2.06	284.0	0.83	0.59
REPA-XL/2 Yu et al. (2025)	800	250×2	675M+86M	1.42	305.7	0.80	0.64
Pixel-space diffusion models							
ADM Dhariwal & Nichol (2021)	400	250	554M	4.59	186.7	0.82	0.52
PixelFlow-XL/4 Chen et al. (2025c)	320	120×2	677M	1.98	282.1	0.81	0.60
PixNerd-XL/16 Wang et al. (2026a)	320	100×2	700M	1.95	300.0	0.80	0.60
JiT-H/16 Li & He (2026)	600	50×2	953M	1.86	303.4	–	–
DeCo-XL/16 Ma et al. (2026a)	600	100×2	682M	1.69	304.0	0.79	0.63
PixelDiT-XL Yu et al. (2026)	320	100×2	797M	1.61	292.7	–	0.64
PixelU-H/16 Guo et al. (2026)	600	50×2	1170M	1.63	305.9	–	–
PatchBridge-H/16	600	50×2	1149M	1.44	317.6	0.81	0.65



Figure 3: Qualitative comparison of JiT and PatchBridge on ImageNet 256×256.

for their shift–scale modulation pairs. Conditioning on (t, y) and the backbone's internal condition-token updates are implicit.

At macro stage m, semantic block $\mathcal{S}_m$ first updates s, after which μ_s is computed once and reused by every structural block in $\mathcal{I}_m$. For $\ell \geq \ell_{\text{det}}$, structural block $\mathcal{B}_\ell$ is followed by detail block $j = \ell - \ell_{\text{det}} + 1$. The first such step initializes d using Eq. (3); later steps reuse the persistent detail state. Each detail update receives modulation from the current structural state and, whenever another structural block remains, returns pooled feedback before that next block.

At a boundary where both reverse paths are active, detail-to-structural feedback precedes structural-to-semantic feedback, allowing the semantic update to include the latest local correction. Both reverse paths are omitted after their parent's final block. Once all macro stages finish, the readout of Sec. 3.4 converts d into overlapping RGB fields and returns their coverage-normalized aggregate $\widehat{\mathbf{x}}$.

3.6 Training Objective

We train the clean-image prediction $\widehat{\mathbf{x}}$ with stabilized velocity matching. With $\delta(t) = \max(1-t, \varepsilon)$ for fixed $\varepsilon > 0$, we define:

$$\mathbf{v}_{\text{tar}} = \frac{\mathbf{x} - \mathbf{x}_t}{\delta(t)}, \qquad \widehat{\mathbf{v}}_\theta = \frac{\widehat{\mathbf{x}} - \mathbf{x}_t}{\delta(t)}, \qquad \mathcal{L}_{\text{diff}} = \mathbb{E}[\text{MSE}(\widehat{\mathbf{v}}_\theta, \mathbf{v}_{\text{tar}})]\,. \tag{8}$$

Here MSE averages squared errors over all pixels and channels. The expectation is over images, noise, timesteps, and conditions. At inference, we integrate $\widehat{\mathbf{v}}_\theta$ from Gaussian noise toward $t = 1$.

4 Experiment

4.1 Experimental Setup

Datasets and evaluation. We evaluate class-conditional generation on ImageNet-1K Deng et al. (2009) at 256×256 and 512×512. Training images are resized and center-cropped to the target resolution, randomly flipped horizontally, and normalized to $[-1, 1]$. Following JiT Li & He (2026), we report FID Heusel et al. (2017), Inception Score Salimans et al. (2016), and precision

and recall Kynkäänniemi et al. (2019) on 50,000 samples. For text-to-image generation, we follow MiniT2I Wang et al. (2026b): pretraining on LLaVA-recaptioned CC12M Changpinyo et al. (2021) and fine-tuning on its roughly 120K-pair mixture of BLIP3o-60K Chen et al. (2025a), DALL-E 3, and ShareGPT-4o-Image Chen et al. (2025b). This relatively modest public-data setting is evaluated with GenEval Ghosh et al. (2023) and DPG-Bench Hu et al. (2024).

Model configurations. PatchBridge-B, L, and H use the corresponding JiT backbones, with semantic, structural, and detail patch sizes of 32, 16, and 8, respectively, at 256×256. For 512×512, we retain these patch sizes and fine-tune PatchBridge-H/16 from its 256×256 checkpoint for 100k steps, regenerating fixed positional encodings for the enlarged token grids. Table 3 lists the depths, stream widths, interaction ranks, and parameter counts.

Training and sampling. Unless otherwise specified, L and H models train for 600 epochs and B models for 800 epochs. We use AdamW with a base learning rate of 5×10^{-5}, five warmup epochs, and no weight decay. Following JiT, timesteps follow a logit-normal distribution with $P_{\text{mean}} = -0.8$ and $P_{\text{std}} = 0.8$, and condition dropout is 0.1. Unless explicitly ablated, we use REPA Yu et al. (2025) with a frozen DINOv2 ViT-B/14 teacher and an alignment coefficient of 0.5, aligning structural blocks 4, 8, and 10 for B, L, and H, respectively. Evaluation uses exponential moving average parameters and classifier-free guidance, with 50 sampling steps for class-conditional generation and 100 for text-to-image generation.

4.2 Class-Conditional Image Generation

Comparison with state-of-the-art methods.

Tables 1 and 2 compare PatchBridge with representative pixel-space generation methods, with latent-space models included for reference. PatchBridge achieves the lowest FID and highest Inception Score among the compared pixel-space models at both resolutions. Although these comparisons span different model sizes and training settings, the gains are also evident at similar parameter counts: at 256×256, PatchBridge-H achieves 1.44 FID with 1149M parameters, compared with 1.63 FID for the 1170M PixelU-H/16, which also builds on JiT. At this resolution, precision and recall further indicate strong sample fidelity and distribution coverage. PatchBridge also achieves 1.54 FID at 512×512, demonstrating competitive generation quality at higher resolution. These results establish its empirical competitiveness across resolutions; the following scaling experiments, component ablations, and capacity-matched comparison further examine the contributions of the proposed design.

Table 2: Class-conditional ImageNet 512×512 generation with classifier-free guidance. Sampling reports steps for JiT and PatchBridge and NFE for other methods.

Method	Sampling	Params	FID↓ w/ CFG	IS↑
Latent-space diffusion models				
DiT-XL/2 Peebles & Xie (2023)	250×2	675M+86M	3.04	240.8
SiT-XL/2 Ma et al. (2024)	250×2	675M+86M	2.62	252.2
REPA-XL/2 Yu et al. (2025)	250×2	675M+86M	2.08	274.6
Pixel-space diffusion models				
ADM-G Dhariwal & Nichol (2021)	250	554M	7.72	172.7
Simple Diffusion Hoogeboom et al. (2023)	250×2	2.0B	3.54	205.0
PixNerd-XL/16 Wang et al. (2026a)	100×2	700M	2.84	245.6
DeCo-XL/16 Ma et al. (2026a)	100×2	682M	2.22	290.0
JiT-H/32 Li & He (2026)	50×2	956M	1.94	309.1
PixelDiT-XL Yu et al. (2026)	100×2	797M	1.81	278.6
PixelU-H/32 Guo et al. (2026)	50×2	1170M	1.92	–
PatchBridge-H	50×2	1152M	1.54	322.3

Scaling across model sizes. Table 3 shows consistent quality gains across B, L, and H configurations, each retaining its JiT structural backbone. For a fair comparison, we retrain JiT-B for 800 epochs to match PatchBridge-B's training budget. PatchBridge reduces FID from 3.68 to 2.75 for B, from 2.36 to 1.76 for L, and from 1.86 to 1.44 for H, with higher Inception Scores at every scale. These gains across model capacities suggest that role-specialized patch processing complements conventional backbone scaling.

Qualitative comparison. Fig. 3 shows class-matched JiT and PatchBridge samples across diverse ImageNet categories. PatchBridge produces recognizable subjects with coherent structure and clear

Table 3: Scaling on ImageNet 256×256. FID changes are relative to the corresponding JiT baselines.

Method	Depth			Width			d_r	Params	FID↓ (Δ)	IS↑
	M	L	K	d_{sem}	d_{str}	d_{det}				
JiT-B/16	–	12	–	–	768	–	–	131M	3.68	273.9
PatchBridge-B/16	4	12	4	384	768	384	64	160M	2.75 (−0.93)	293.9
JiT-L/16	–	24	–	–	1024	–	–	459M	2.36	298.5
PatchBridge-L/16	8	24	8	512	1024	512	96	560M	1.76 (−0.60)	310.8
JiT-H/16	–	32	–	–	1280	–	–	953M	1.86	303.4
PatchBridge-H/16	10	32	10	640	1280	640	128	1149M	1.44 (−0.42)	317.6

Table 5: Component ablation of the B model on ImageNet 256×256.

Semantic stream	Detail stream	Overlapping readout	REPA	FID↓	IS↑
✗	✗	✗	✗	3.68	273.9
✗	✗	✗	✓	3.49	271.3
✓	✗	✗	✗	3.44	277.6
✓	✓	✗	✗	3.06	284.7
✓	✓	✓	✗	2.91	298.1
✓	✓	✓	✓	2.75	293.9

local details. The examples span object-centered images and wider scenes, illustrating detail and composition at different spatial scales.

4.3 Text-to-Image Generation

We extend PatchBridge to text-to-image generation within the MiniT2I framework Wang et al. (2026b). Despite using a limited training dataset, PatchBridge-B/32 achieves 0.87 on GenEval and 84.1 on DPG-Bench (Table 4). It achieves the highest GenEval and DPG-Bench scores among the compared pixel-space models, demonstrating strong compositional alignment under a limited-data training regime. These results use only 289M generator parameters and a 341M text encoder, substantially fewer generator parameters than the billion-parameter baselines. Relative to MiniT2I-B/32, PatchBridge improves GenEval from 0.82 to 0.87 and DPG-Bench from 82.3 to 84.1, with only 29M additional generator parameters and the same text encoder. The strong performance with modest model capacity supports the effectiveness of role-specialized patch processing beyond class-conditional generation.

Table 4: Pixel-space text-to-image results.

Model	Params	GenEval↑	DPG↑
PixelFlow-XL/4 Chen et al. (2025c)	882+800M	0.60	77.9
PixelDiT-T2I Yu et al. (2026)	1.3+2.6B	0.78	83.7
PixNerd-XXL/16 Wang et al. (2026a)	1.2+1.7B	0.73	80.9
PixelGen-XXL/16 Ma et al. (2026b)	1.1+1.7B	0.79	–
DeCo-XXL/16 Ma et al. (2026a)	1.1+1.7B	0.86	81.4
MiniT2I-B/32 Wang et al. (2026b)	260+341M	0.82	82.3
PatchBridge-B/32	289+341M	0.87	84.1

Table 6: Reciprocal feedback on ImageNet 256×256.

Str.→Sem.	Det.→Str.	FID↓	IS↑
✗	✗	3.97	256.4
✓	✗	3.86	259.6
✓	✓	3.51	263.9

Table 7: Overlap extent.

q/p_{det}	q	Overlap per side	FID↓	IS↑
1.00	8	0	3.84	256.6
1.25	10	1	3.80	260.1
1.50	12	2	3.51	263.9
2.00	16	4	3.69	261.4

4.4 Ablation Studies

Component contributions. Table 5 progressively adds PatchBridge components to JiT-B. All variants train for 800 epochs with identical optimization, data processing, sampling, and evaluation settings. Adding the semantic stream improves FID from 3.68 to 3.44 and IS from 273.9 to 277.6. Activating the detail stream in later backbone stages yields a larger gain, reducing FID to 3.06 and increasing IS to 284.7. Overlapping readout further improves FID to 2.91 and IS to 298.1. Thus, without auxiliary representation supervision, PatchBridge improves FID by 0.77 over JiT-B. Adding REPA reduces FID to 2.75, an overall improvement of 0.93, although IS decreases to 293.9 while remaining above the baseline. These results support the effectiveness of the architecture itself and show that representation alignment provides additional FID gains.

Number of streams. We test whether a finer scale benefits PatchBridge-B by adding a fourth stream with 4×4 patches and the same adjacent-stream interaction design. Under identical 200-epoch settings, the four-stream model yields an FID of 3.53, compared with 3.51 for the three-stream model. Despite comparable quality, the fine-scale stream substantially increases computational cost, making the three-stream design a better quality–cost trade-off.

Reciprocal cross-scale feedback. We isolate the contribution of child-to-parent feedback by retaining both parent-to-child modulation pathways and progressively enabling the reverse connec-

tions. All variants use the same 200-epoch training budget and identical training and evaluation settings. As shown in Table 6, structural-to-semantic feedback reduces FID from 3.97 to 3.86, and additionally enabling detail-to-structural feedback further reduces it to 3.51, with consistent improvements in IS. These results support complementing top-down conditioning with bottom-up feedback, allowing finer-scale states to inform subsequent coarser-scale processing.

Overlap extent. The overlapping readout allows neighboring detail tokens to jointly contribute to shared output pixels, complementing the scale-specialized processing with collaborative image reconstruction. We vary the prediction size q while fixing the detail-token stride to $p_{det} = 8$, using identical training and evaluation settings for 200 epochs. As shown in Table 7, moderate overlap improves generation quality over the disjoint readout: increasing q from 8 to 12 reduces FID from 3.84 to 3.51 and improves IS from 256.6 to 263.9. Further increasing q to 16 worsens FID to 3.69, indicating that larger prediction fields do not necessarily yield additional gains. We therefore adopt $q = 12$, which achieves the best FID and IS among the tested configurations.

4.5 Quality and Efficiency

Patch-boundary analysis. Fig. 4 compares local image continuity between H-sized JiT and PatchBridge models using the same class label and sampling settings. Because the models learn distinct generative mappings, matched conditions do not imply pixel-aligned samples; the figure therefore provides a qualitative comparison of boundary continuity. The enlarged crop around the bird–background interface reveals visible grid-like discontinuities in the JiT output, whereas PatchBridge produces a smoother object contour with more coherent color and texture transitions. This aligns with our overlapping readout, which aggregates neighboring detail-token predictions at shared pixels with coverage normalization.

Table 8: Capacity and inference cost. Both Deeper JiT-B and PatchBridge-B use REPA.

Model	Depth	Params (M)	FID↓	GFLOPs↓	Peak GB↓	img/s↑
JiT-B	12	131	3.68	50.45	1.95	7.87
Deeper JiT-B	15	163	3.30	63.46	2.08	6.19
PatchBridge-B	12	160	2.75	60.90	2.15	6.53

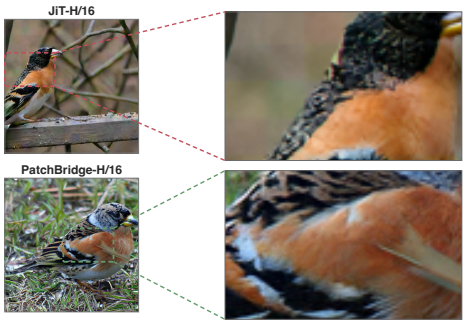


Figure 4: Patch-boundary comparison.

Capacity-matched comparison. Compared with JiT-B, PatchBridge-B increases parameters from 131M to 160M and GFLOPs from 50.45 to 60.90, yet substantially improves FID from 3.68 to 2.75 (Table 8). For a capacity-matched comparison, we deepen JiT-B from 12 to 15 blocks and train both Deeper JiT-B and PatchBridge-B with REPA under the same 800-epoch training and evaluation protocol. Deeper JiT-B achieves 3.30 FID with 163M parameters and 63.46 GFLOPs, versus 2.75 FID with 160M parameters and 60.90 GFLOPs for PatchBridge-B. Measured on a single A100 GPU with batch size 128, PatchBridge-B achieves higher throughput (6.53 vs. 6.19 images/s), with slightly higher peak memory usage (2.15 vs. 2.08 GB). These gains at comparable capacity and compute support the effectiveness of PatchBridge's architectural design.

5 Conclusion

We introduced PatchBridge to address patch-role coupling in pixel-space diffusion Transformers. Its semantic, structural, and detail streams use role-specific patch granularities and lifetimes, while reciprocal cross-scale interactions keep their denoising states coordinated. An overlapping clean-image readout further lets neighboring detail tokens contribute to shared pixels. PatchBridge achieves FIDs of 1.44 and 1.54 on class-conditional ImageNet at 256×256 and 512×512, respectively, and scores 0.87 on GenEval and 84.1 on DPG-Bench in text-to-image generation. Component ablations and a capacity-matched JiT comparison support the value of this design beyond increasing model size alone. These results suggest that allocating patch granularity and computation according to representational role is an effective approach to direct pixel-space generation.

AI USE STATEMENT

We used generative AI tools only to improve the clarity and fluency of the manuscript and to assist with literature search and review. The authors checked the AI-assisted edits and independently verified the cited literature against the original sources. Generative AI tools were not used to develop the research ideas or methods, design experiments, implement the method, analyze data, or interpret results. The authors take full responsibility for the paper's content.

REFERENCES

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. BLIP3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv:2505.09568*, 2025a.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. ShareGPT-4o-Image: Aligning multimodal models with GPT-4o-level image generation. *arXiv:2506.18095*, 2025b.

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

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

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

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 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. In *ECCV*, 2026.

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

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

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

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *CVPR*, 2025.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv:2403.05135*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

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

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 *ICCV*, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, 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 *ECCV*, 2024.

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

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv:2602.02493*, 2026b.

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

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

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

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

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

Xianbang Wang, Hanhong Zhao, Yiyang Lu, Kangyang Zhou, Linrui Ma, and Kaiming He. Minit2i: A minimalist baseline for text-to-image generation, 2026b. URL `https://peppaking8.github.io/#/post/minit2i`.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 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 *ICLR*, 2025.

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

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