CORE: LEARNING CONTEXTUALLY GROUNDED REPRESENTATIONS FOR PIXEL-SPACE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent self-distillation methods improve latent-space diffusion by aligning noisy representations with cleaner targets, often derived from an exponential moving average (EMA) teacher. However, these gains do not consistently transfer to pixel-space generation, where models must learn spatial structure and fine-grained appearance directly from noisy pixels. We find that high feature similarity can persist even when spatial correspondence deteriorates sharply, revealing that representation alignment alone does not ensure utility for pixel prediction. To address this limitation, we introduce **CORE**, a spatial self-supervision framework for learning contextually grounded representations in pixel-space diffusion. *Correlated Dual Observations* provide paired noisy and cleaner views of the same image with correlated noise for training a shared generator. Built on these observations, *Contextual Representation Grounding* uses two coupled prediction tasks on a shared representation at masked locations. One predicts cleaner features from visible context, while the other predicts correction targets formed by selecting and scaling cleaner-induced prediction changes according to ground-truth pixel residuals. Together, these tasks encourage contextual recoverability while linking representation learning to pixel prediction errors. CORE requires neither an external representation encoder nor an independent EMA teacher and leaves inference unchanged. Experiments on ImageNet at 256×256 and 512×512 demonstrate that CORE achieves consistently lower FID than the evaluated self-supervision baselines. At 600 epochs, CORE-L/16 even achieves a lower FID than PixelREPA, which uses external representation supervision. Code will be publicly released.

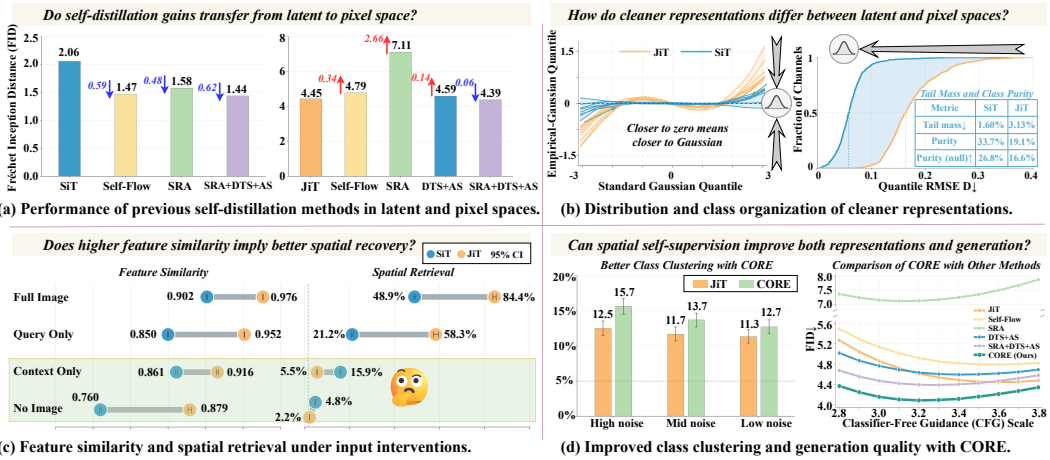
1 INTRODUCTION

Learning useful representations is important for diffusion-based image generation. Representation alignment methods such as REPA (Yu et al., 2025) demonstrate that guidance from pretrained visual encoders can improve both training efficiency and generation quality. More recent methods obtain representation supervision from the generative model itself. SRA (Jiang et al., 2026b) and Self-Flow (Chefer et al., 2026) align noisy features with cleaner features from an exponential moving average (EMA) teacher, reducing reliance on external representation encoders. Together with advances in noise scheduling and attention separation (Jiang et al., 2026c), these methods establish self-distillation as a promising approach to representation learning for latent-space generation.

Whether these benefits extend to pixel-space diffusion remains less clear. Recent models such as JiT (Li & He, 2026) demonstrate that transformers can generate images directly in pixel space without a pretrained tokenizer. This setting presents a different representation learning problem, as the model must learn spatial structure and fine-grained detail directly from noisy pixels (Li & He, 2026; lei et al., 2026; Wang et al., 2026a). Our evaluations reveal that methods with substantial reported improvements in latent space do not consistently improve JiT under the evaluated training settings, as summarized in Figure 1(a). These motivate revisiting self-supervision for pixel-space generation.

To understand this transfer gap, we compare the distribution and organization of representations in latent and pixel spaces. Latent diffusion models operate on perceptually compressed representations produced by pretrained autoencoders (Rombach et al., 2022). In the VAE setting, compression and regularization toward a Gaussian prior provide a more compact modeling space, while the decoder handles the final reconstruction of pixels (Kingma & Welling, 2013; Rombach et al., 2022).

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.



(a) Performance of previous self-distillation methods in latent and pixel spaces.

(b) Distribution and class organization of cleaner representations.

(c) Feature similarity and spatial retrieval under input interventions.

(d) Improved class clustering and generation quality with CORE.

Figure 1: Motivating contextually grounded representations for pixel-space diffusion. (a) Self-distillation gains in latent space do not consistently transfer to pixel space. (b) SiT cleaner features exhibit more Gaussian-like channel marginals and higher class neighborhood purity than JiT. (c) Removing image content degrades spatial retrieval despite high feature similarity. (d) CORE improves class neighborhood purity across noise levels and achieves lower FID than competing methods. Implementation details and analysis protocols are provided in Appendix A.5.

Pixel-space models do not inherit this division of labor and must learn spatial structure, local appearance, and fine-grained detail within the denoising network. As shown in Figure 1(b), the examined cleaner features of SiT (Ma et al., 2024) exhibit more Gaussian-like channel marginals and higher class neighborhood purity than those of JiT. These differences motivate reassessing how cleaner representations should be used as supervision.

For pixel-space generation, capturing spatial structure and local detail is important for accurate pixel prediction (Yu et al., 2026; Wang et al., 2026c). When using cleaner features as supervision targets, a key question is whether high feature agreement also reflects accurate spatial correspondence. With fixed backbones, we train linear predictors on unmasked inputs and test whether predicted features identify their matching locations in the cleaner feature map. Removing content around the queried location leaves JiT's cosine similarity at 0.916, while retrieval accuracy drops from 84.4% to 5.5% (Figure 1(c)). Under the same intervention, SiT exhibits lower similarity (0.861) but higher retrieval (15.9%). Even without image content, both models retain high similarity despite weak retrieval. Thus, feature agreement alone does not establish spatial correspondence.

These findings suggest that pixel-space self-supervision should address not only feature alignment, but also spatial relationships between image regions. We therefore mask target regions and train the model to predict their features from visible context (Chen et al., 2022; Assran et al., 2023). With local content withheld, this task encourages the model to use information across regions. However, contextual feature prediction does not directly supervise corrections to pixel prediction errors. We thus complement contextual prediction with error-related supervision and ask *how to learn representations grounded in visible spatial context and useful for accurate pixel prediction*.

To this end, we introduce CORE, which combines Correlated Dual Observations (CDO) with Contextual Representation Grounding (CRG). CDO pairs noisy and cleaner views of the same image with correlated noise and trains both through a shared generator. The resulting features and image predictions also provide target sources for CRG. To ground representations in visible context, CRG hides selected patch tokens before the shared early layers and jointly trains two prediction heads on the resulting representation. Contextual Feature Prediction (CFP) predicts cleaner features at the hidden locations, providing explicit feature-level supervision for the missing regions. Pixel-Guided Correction (PGC) complements this supervision with correction targets tied to the generator's pixel prediction errors. These targets are formed by selecting and scaling differences between noisy and cleaner image predictions according to the residual between the base prediction and ground truth. CFP and PGC thus jointly supervise what to infer about missing regions and how that information relates to current prediction errors. We stop gradients through both targets, while the shared generator remains trainable without an independent EMA teacher. At inference, CORE preserves the original generator architecture and sampling procedure with no additional computational overhead.

Experiments demonstrate that CORE improves both representation quality and generation performance. The left panel of Figure 1(d) shows higher class neighborhood purity than JiT across noise levels along the standard generation path, indicating better class organization. The right panel shows lower FID than the evaluated self-supervision baselines across the tested guidance scales.

Our contributions are summarized as follows:

- We show that latent-space self-distillation gains do not consistently transfer to pixel-space diffusion. Representation analyses and input interventions reveal that high feature similarity can coexist with poor spatial correspondence, motivating supervision beyond alignment.
- We introduce CORE, combining paired generative training using CDO with CRG. Within CRG, CFP and PGC jointly supervise a shared contextual representation with missing-region feature and pixel-error-guided correction targets, without an external representation encoder or an independent EMA teacher, while leaving inference unchanged.
- We demonstrate improved generation quality on ImageNet over the evaluated self-supervision baselines, with component ablations supporting the joint design.

2 Related Work

2.1 Pixel-Space Diffusion Models

Early diffusion models generate images directly in pixel space (Ho et al., 2020; Song et al., 2021; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Karras et al., 2022), whereas latent-space models operate on encoded image representations (Rombach et al., 2022; Bao et al., 2022; Peebles & Xie, 2023; Ma et al., 2024; Esser et al., 2024; Wang et al., 2026d; Leng et al., 2025; Zheng et al., 2026; Shi et al., 2025; Kouzelis et al., 2025). Recent work revisits direct pixel-space generation through improved training strategies and architectures. Simple Diffusion (Hoogeboom et al., 2023) and SiD2 (Hoogeboom et al., 2025) develop high-resolution training recipes, while PixelFlow (Chen et al., 2025) adopts cascaded flow modeling. Architectural advances include neural-field decoding in PixNerd (Wang et al., 2026c), patch-level modeling and pixel-level refinement in PixelDiT (Yu et al., 2026) and DiP (Chen et al., 2026), and frequency-decoupled generation in DeCo (Ma et al., 2026). JiT (Li & He, 2026) demonstrates competitive generation with large-patch Transformers through clean-image prediction. Complementary training approaches investigate self-supervised pretraining (lei et al., 2026), latent-to-pixel transfer (Jiang et al., 2026a), representation alignment through PixelREPA (Shin et al., 2026), and joint latent–pixel modeling through Latent Forcing (Baade et al., 2026). Pixel-space generation also extends to unified models, including HiDream-O1-Image (Cai et al., 2026), SenseNova-U1 (Diao et al., 2026b), SenseNova-U1.5 (Diao et al., 2026a), Tuna-2 (Liu et al., 2026), and Representation Forcing (Wang et al., 2026e). Our work builds on JiT and focuses on spatial self-supervision during generative training, preserving its original inference architecture and sampling procedure.

2.2 Representation Learning and Self-Distillation for Diffusion

Representation learning supports generation through latent-space design, explicit representation modeling, and feature supervision. VA-VAE (Yao et al., 2025), REPA-E (Leng et al., 2025), RAE (Zheng et al., 2026), RAEv2 (Singh et al., 2026b), and PAE (Yue et al., 2026) improve generative latent spaces through alignment, tokenizer tuning, or representation-based encoding. RCG (Li et al., 2024) conditions generation on sampled representations, while ReDi (Kouzelis et al., 2025) jointly models image latents and semantic features. Latent Forcing (Baade et al., 2026) and Representation Forcing (Wang et al., 2026e) further use intermediate representations to guide pixel generation. WiT (Wang et al., 2026a) predicts semantic waypoints from noisy pixels and uses them to condition pixel-space generation, with waypoint targets derived from pretrained visual representations. For feature supervision, REPA (Yu et al., 2025) aligns diffusion features with pretrained visual representations, iREPA (Singh et al., 2026a) emphasizes target spatial structure, and PixelREPA (Shin et al., 2026) adapts alignment to pixel-space models. DeepFlow (Shin et al., 2025) introduces intermediate velocity supervision, while SRA 2 (Wang et al., 2026b) reuses pretrained VAE features as alignment targets. Without external representation encoders, SRA (Jiang et al., 2026b) aligns earlier, higher-noise features with deeper, lower-noise EMA teacher features. Self-

Flow (Chefer et al., 2026) combines internal distillation with token-wise dual-timestep scheduling, and Attention Separation (Jiang et al., 2026c) examines the contribution of heterogeneous-noise augmentation by blocking cross-group attention. CORE builds on CDO and introduces CRG, which jointly supervises a contextual representation through CFP and PGC. Its feature and correction targets are constructed from the jointly trained generator and ground truth, without an external representation encoder or an independent EMA teacher.

2.3 Masked Modeling and Contextual Prediction

Masked modeling learns representations by predicting missing information from visible context. Context Encoders (Pathak et al., 2016) and MAE (He et al., 2022) reconstruct image pixels, while BEiT (Bao et al., 2021), MaskFeat (Wei et al., 2022), and iBOT (Zhou et al., 2021) explore discrete tokens, handcrafted features, and online teacher targets, respectively. CAE (Chen et al., 2022) and I-JEPA (Assran et al., 2023) predict target-region representations from visible context, and MAGE (Li et al., 2023) and ADDP (Tian et al., 2023) connect masked modeling with image generation. Within diffusion, DiffMAE (Wei et al., 2023) uses conditional denoising for visual pretraining and inpainting. MaskDiT (Zheng et al., 2023) combines visible-patch denoising with masked-patch reconstruction, while MDT (Gao et al., 2023) uses masked latent modeling to strengthen contextual learning. MC-DiT (Zheng et al., 2024) addresses limitations of noisy masked reconstruction through clean-to-clean contextual learning, and SD-DiT (Zhu et al., 2024) combines teacher–student discrimination with diffusion training. CORE incorporates CRG into the shared early layers of a pixel-space generator. Within CRG, CFP predicts cleaner features at masked locations, while PGC predicts correction targets selected and scaled according to ground-truth pixel residuals. Both tasks supervise the same contextual representation, connecting missing-region feature learning to the prediction errors.

3 Methodology

We introduce CORE, a spatial self-supervision framework built on CDO and CRG. CDO provides paired noisy and cleaner observations for generative training and target construction. CRG grounds a shared contextual representation through two complementary tasks, with CFP providing cleaner-feature supervision for missing regions and PGC supplying correction targets tied to pixel prediction errors. Figure 2 summarizes the pipeline, and Algorithm 1 details the training procedure.

3.1 Correlated Dual Observations (CDO)

Let $x_0 \in \mathbb{R}^{C\times H\times W}$ be a clean image and c its class condition. Given a timestep $t \sim p(t)$ and Gaussian noise $\epsilon_1 \sim \mathcal{N}(0, \sigma^2 I)$, the noisy observation and base prediction are

$$x_t = tx_0 + (1-t)\epsilon_1, \qquad y_b = f_\theta(x_t, t, c). \tag{1}$$

Under this convention, larger t corresponds to a cleaner observation. The timestep distribution and noise scale follow the underlying generator, such as JiT (Li & He, 2026).

To construct a paired cleaner observation, we independently sample $\epsilon_2 \sim \mathcal{N}(0, \sigma^2 I)$ and define

$$\epsilon^+ = \frac{\epsilon_1 + \epsilon_2}{\sqrt{2}}, \qquad t^+ = \frac{t}{t + (1-t)/\sqrt{2}}, \qquad x_t^+ = t^+ x_0 + (1-t^+)\epsilon^+. \tag{2}$$

We obtain the cleaner view by averaging two independently corrupted observations at t and rescaling the result. This determines t^+ directly from t and combines the base noise with fresh noise, so the two views share some noise without using identical perturbations. The fused noise retains covariance $\sigma^2 I$, while $t^+ > t$ for $0 < t < 1$. The generator uses the same parameters θ for both views, producing $y_c = f_\theta(x_t^+, t^+, c)$. Both observations use the same class condition, including the same dropped condition when classifier-free training is applied.

Following JiT (Li & He, 2026), we train both branches via flow matching with x-prediction and v-loss. For a noisy observation x_u at timestep u and its clean-image prediction y, we define

$$\ell_{\text{gen}}(y, x_0; u) = \|\widehat{v}_u - v_u\|_2^2, \quad \widehat{v}_u = \frac{y - x_u}{\max(1-u, \tau)}, \quad v_u = \frac{x_0 - x_u}{\max(1-u, \tau)}. \tag{3}$$

Here, $\tau > 0$ is a numerical floor. The generator predicts clean images, which are converted to velocities for loss computation.

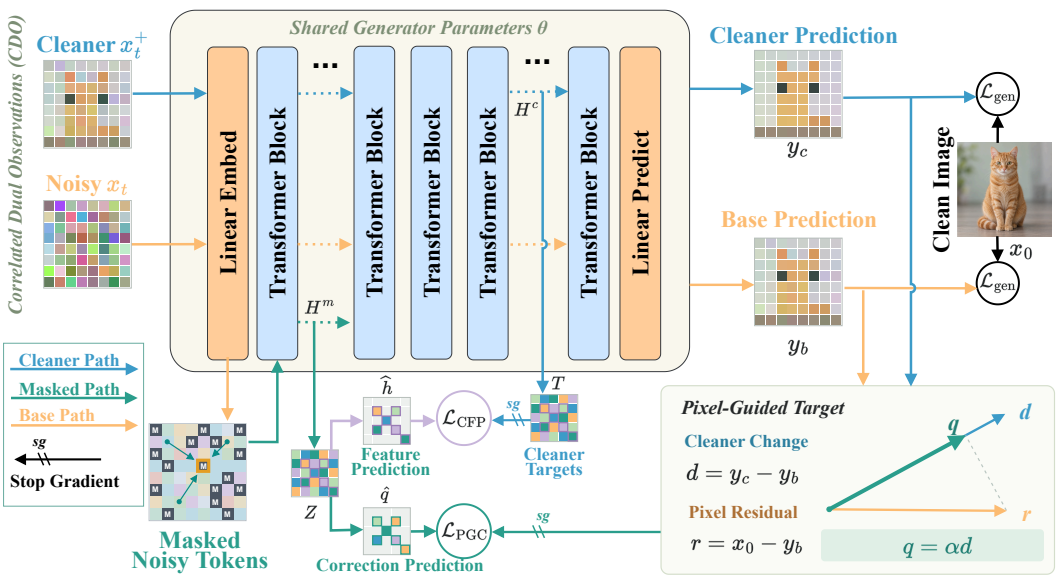


Figure 2: Overview of CORE. CDO provides noisy and cleaner observations for paired generative training with shared parameters. CRG replaces selected patch embeddings with mask tokens and uses shared early Transformer blocks to form a contextual representation. Two readouts jointly supervise this representation through CFP and PGC. Cleaner features provide CFP targets, while paired image predictions and ground truth determine PGC targets. Colored paths denote separate evaluations with shared parameters, and sg denotes stop-gradient.

3.2 Contextual Representation Grounding (CRG)

CRG couples two goals on a shared contextual representation. CFP targets *contextual recoverability* by predicting cleaner features at hidden locations, while PGC targets *corrective utility* through correction targets tied to pixel prediction errors. Both share the masked input, early generator blocks, and bottleneck. Let $E_\theta(x_t) \in \mathbb{R}^{N \times D_h}$ denote the noisy image's patch embeddings. For each image, we randomly select a fraction of positions $\mathcal{M} \subset \{1, \ldots, N\}$ and denote the remaining visible positions by $\mathcal{V}$. Masking is applied after patch embedding and before spatial attention:

$$\widetilde{e}_i = \begin{cases} [E_\theta(x_t)]_i + p_i, & i \in \mathcal{V}, \\ m + p_i, & i \in \mathcal{M}, \end{cases} \tag{4}$$

where m is a learned mask token and p_i denotes positional information. Thus, replacing a token removes its local image content while retaining its spatial location.

The masked sequence is processed by the first K Transformer blocks of the same generator. Within these blocks, visible queries attend only to visible keys and values, whereas masked queries can attend to the complete post-mask sequence. Equivalently, the additive attention mask is

$$A_{ij} = \begin{cases} -\infty, & i \in \mathcal{V} \text{ and } j \in \mathcal{M}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Visible states remain independent of masked-token states. With their content removed, masked locations gather information from visible patches and class, timestep, and positional conditioning.

Let H^m be the resulting early-layer features. A shared token-wise bottleneck produces the spatial representation Z, and two linear readouts predict cleaner features and pixel corrections:

$$Z_i = \mathrm{GELU}(W_p\,\mathrm{LN}(H_i^m) + b_p)\,, \qquad \widehat{h}_i = W_h Z_i + b_h, \qquad \widehat{q}_i = W_q Z_i + b_q. \tag{6}$$

Here LN is token-wise layer normalization without learned affine parameters. The projection compresses channels but preserves every spatial position. Since the readouts do not mix tokens, any use of neighboring image content must occur within the shared Transformer blocks.

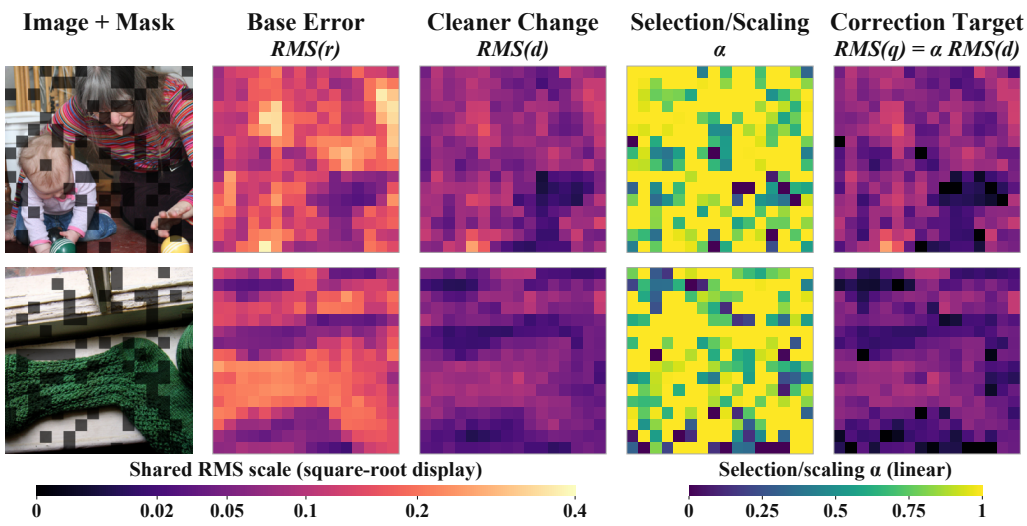


Figure 3: Pixel-error-guided correction targets. From left to right, we show the image with masked regions, patchwise RMS maps of the base residual $r = x_0 - y_b$ and cleaner-induced change $d = y_c - y_b$, the selection and scaling coefficient α, and the RMS map of the target $q = \alpha d$. Residual-guided selection and scaling suppress misaligned changes and limit correction magnitudes.

3.2.1 Contextual Feature Prediction (CFP)

CFP provides explicit feature-level supervision for the missing regions. Let H^c denote layer-ℓ_c image-token features from the complete cleaner forward pass.

The feature target is $T_i = \mathrm{sg}(\mathrm{LN}(H_i^c))$. Writing $\overline{h} = h/\max(\|h\|_2, \varepsilon_f)$ for stabilized ℓ_2 normalization of a feature vector h, the contextual objective is

$$\mathcal{L}_{\mathrm{CFP}} = \left\langle 1 - \left\langle \overline{\widehat{h}_i}, \overline{T_i} \right\rangle \right\rangle_{\mathcal{M}}, \tag{7}$$

where $\langle\cdot\rangle_{\mathcal{M}}$ denotes the empirical average over all masked positions in the current minibatch. The target comes from the complete cleaner observation, whereas its predictor has access only to the masked noisy observation and the conditioning information.

3.2.2 Pixel-Guided Correction (PGC)

CFP supervises missing-region features but does not directly constrain image corrections. PGC complements it with correction targets constructed from paired image predictions and ground truth. Both readouts share Z, jointly supervising feature and correction prediction from visible context. For each patch i, let $\mathcal{P}_i$ extract its RGB pixel vector. We define the base residual and cleaner-induced candidate change as

$$r_i = \mathrm{sg}(\mathcal{P}_i(x_0 - y_b)), \qquad d_i = \mathrm{sg}(\mathcal{P}_i(y_c - y_b)). \tag{8}$$

The candidate d_i is not assumed to improve the base prediction everywhere. We select and scale it according to its agreement with the true residual:

$$\alpha_i = \mathrm{clip}_{[0,1]}\left(\frac{\langle r_i, d_i\rangle}{\|d_i\|_2^2 + \varepsilon_q}\right), \qquad q_i = \mathrm{sg}(\alpha_i d_i). \tag{9}$$

The inner product is computed over the full pixel-patch vector. Changes opposed to the residual are rejected, while useful changes are retained with a bounded scale.

The coefficient α_i minimizes the regularized patchwise pixel prediction error

$$\alpha_i = \arg\min_{a\in[0,1]} \left\{\|r_i - a d_i\|_2^2 + \varepsilon_q a^2\right\}. \tag{10}$$

Since $a = 0$ is feasible, the constructed target satisfies $\|r_i - q_i\|_2^2 \le \|r_i\|_2^2$. Thus, q_i selects a scaled cleaner-induced change that does not increase the base prediction's patchwise pixel error. The

Table 1: Generation results on ImageNet 256×256. Params counts inference-time parameters. Aux. supervision denotes additional representation supervision beyond the generative objective, using internal targets (Self) or an external encoder (External). In this column, – denotes no auxiliary representation supervision. Dashes in metric columns indicate unavailable tabulated values.

Method	Params	Aux. supervision	200 epochs		600 epochs	
			FID↓	IS↑	FID↓	IS↑
B/16						
JiT (Li & He, 2026)	131M	–	4.45	257.1	3.66	275.1
Self-Flow (Chefer et al., 2026)	131M	Self	4.79	243.6	3.95	263.8
SRA (Jiang et al., 2026b)	131M	Self	7.11	195.0	4.25	252.2
DTS+AS (Jiang et al., 2026c)	131M	Self	4.59	273.4	3.91	282.0
SRA+DTS+AS (Jiang et al., 2026c)	131M	Self	4.39	274.5	3.72	284.5
CORE (Ours)	131M	Self	4.09	257.2	3.38	269.4
L/16						
PixelREPA (Shin et al., 2026)	459M	External	–	–	2.11	309.5
JiT (Li & He, 2026)	459M	–	2.79	–	2.36	298.5
SRA+DTS+AS (Jiang et al., 2026c)	459M	Self	2.92	291.4	3.08	305.4
CORE (Ours)	459M	Self	2.51	283.9	2.02	294.0

correction head is trained to predict this target from the shared contextual representation. Figure 3 illustrates how residual-guided selection and scaling produce spatially varying correction targets.

The correction objective is

$$\mathcal{L}_{\mathrm{PGC}} = \frac{\mathrm{MSE}_{\mathcal{M}}(\widehat{q}, q)}{\max(\mathrm{sg}[\mathrm{MSE}_{\mathcal{M}}(r, 0)], \varepsilon_r)}. \tag{11}$$

Here, $\mathrm{MSE}_{\mathcal{M}}$ averages over all RGB patch coordinates at masked positions in the minibatch. Normalization uses the detached minibatch-averaged base error. Correction prediction shapes the shared spatial representation only during training, and neither q nor $\widehat{q}$ is added to the base prediction. The constants $\varepsilon_f, \varepsilon_q, \varepsilon_r > 0$ provide numerical stabilization.

3.3 Training and Inference

The complete objective combines paired generative training using CDO with the two complementary supervision terms of CRG.

$$\mathcal{L}_{\mathrm{CORE}} = \left\langle \ell_{\mathrm{gen}}(y_b, x_0; t) + \eta\, \ell_{\mathrm{gen}}(y_c, x_0; t^+) \right\rangle_{\mathcal{B}} + \lambda_{\mathrm{CFP}} \mathcal{L}_{\mathrm{CFP}} + \lambda_{\mathrm{PGC}} \mathcal{L}_{\mathrm{PGC}}, \tag{12}$$

where $\langle \cdot \rangle_{\mathcal{B}}$ averages over minibatch images. The CFP and PGC objectives within CRG jointly update the early layers, with both targets detached. Both full-image paths are trained through their respective generative losses. For JiT-B/16, we use a 25% mask ratio on the 16×16 token grid, $K = 4$ shared blocks, cleaner features from block $\ell_c = 8$, and a $768 \to 128$ bottleneck with separate linear readouts. We set $\eta = 0.25$ and $\lambda_{\mathrm{CFP}} = \lambda_{\mathrm{PGC}} = 0.05$, and zero-initialize the correction readout. Training requires two full generator evaluations and one shared-prefix evaluation, without an external representation encoder or independent EMA teacher. Masking and spatial prediction heads are unused at inference, leaving the generator and sampling procedure unchanged. The complete training procedure is summarized in Algorithm 1 in Appendix A.1.

4 Experimental Validation

Experimental settings are detailed in Appendix A.2 and A.4. Comparisons of trainable parameters and inference overhead are provided in Appendix A.3.

4.1 Main Result

Quantitative Results Table 1 summarizes the B/16 and L/16 results on ImageNet 256×256. At 200 epochs, CORE-B/16 achieves 4.09 FID, improving over JiT's 4.45 and SRA+DTS+AS's

Table 2: Generation results on ImageNet 512×512 with JiT-B/32.

Method	Params	Aux. supervision	200 epochs FID↓	200 epochs IS↑	600 epochs FID↓	600 epochs IS↑
JiT (Li & He, 2026)	133M	–	4.64	–	4.02	271.0
SRA+DTS+AS (Jiang et al., 2026c)	133M	Self	5.04	279.4	4.28	295.7
CORE (Ours)	133M	Self	4.50	268.7	3.86	281.0

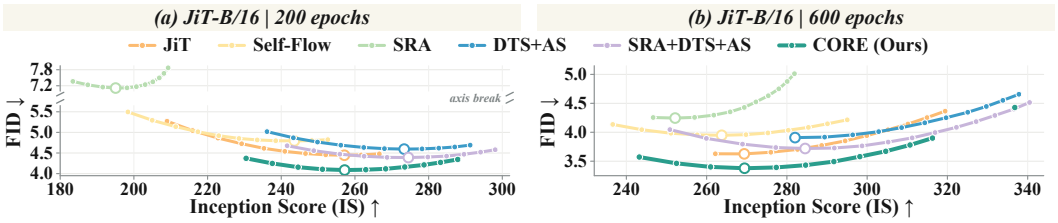


Figure 4: FID–IS tradeoffs of JiT-B/16 on ImageNet 256×256 at 200 and 600 training epochs. Each point represents a CFG setting evaluated with 50,000 samples, and open circles mark the lowest FID for each method. JiT curves use reproduced evaluations.

4.39. Self-Flow and SRA do not improve over JiT, while SRA+DTS+AS provides a smaller gain. CORE also achieves lower FID throughout the tested guidance range (Figure 1(d)), demonstrating consistent improvements across guidance scales.

At 600 epochs, CORE-B/16 reaches an FID of 3.38, improving over JiT's 3.66 and all four reproduced comparison methods. SRA+DTS+AS remains the strongest reproduced self-supervision baseline with an FID of 3.72, which CORE improves by 0.34. We therefore select SRA+DTS+AS as the self-supervision baseline for our larger-model and higher-resolution experiments.

Figure 4 further compares FID and IS across guidance scales. At both training durations, CORE's curve lies below SRA+DTS+AS over a broad overlapping IS range, showing that its FID advantage is not confined to comparisons at independently selected best-FID operating points.

CORE-L/16 achieves FIDs of 2.51 and 2.02 at 200 and 600 epochs, respectively, outperforming JiT and SRA+DTS+AS. SRA+DTS+AS does not improve FID over JiT at either duration, whereas CORE maintains consistent gains. At 600 epochs, CORE even achieves a lower FID than PixelREPA (2.11), which uses external representation supervision. These results show that CORE scales to larger models and can outperform externally supervised alignment in this setting.

Table 2 examines higher-resolution generation with JiT-B/32. CORE achieves FIDs of 4.50 and 3.86 at 200 and 600 epochs, respectively, outperforming both JiT and SRA+DTS+AS at both training durations. These results show that CORE improves higher-resolution generation, maintaining the same inference cost as JiT.

Qualitative Results Figure 5 presents selected ImageNet samples generated by CORE. The examples illustrate coherent object shapes and detailed local appearance, including fur, feathers, and floral structures, alongside varied poses and backgrounds. Sampling uses the ordinary generation path without auxiliary spatial prediction heads. More samples are shown in Appendix A.7.

4.2 Ablation Study

We examine the contributions of CDO and CRG using JiT-B/16 trained for 200 epochs. Table 3 first compares JiT, CDO-only, and the complete CORE framework. CDO-only retains paired generative training without CRG. We then remove CFP or PGC from CORE to examine their roles within CRG. Shared training settings and the weights of the retained objectives remain unchanged.

Paired generative training with CDO reduces FID from 4.45 to 4.17. Adding CRG further improves FID to 4.09, showing an additional benefit beyond paired generative training alone.

Table 3: Ablation of CDO and CRG on ImageNet 256×256 with JiT-B/16 trained for 200 epochs.

Configuration	CDO	CRG		FID↓	IS↑
		CFP	PGC		
JiT (Li & He, 2026)	–	–	–	4.45	257.1
CDO only	✓	–	–	4.17	260.7
CORE w/o CFP	✓	–	✓	4.27	259.4
CORE w/o PGC	✓	✓	–	4.13	257.4
CORE (Ours)	✓	✓	✓	4.09	257.2

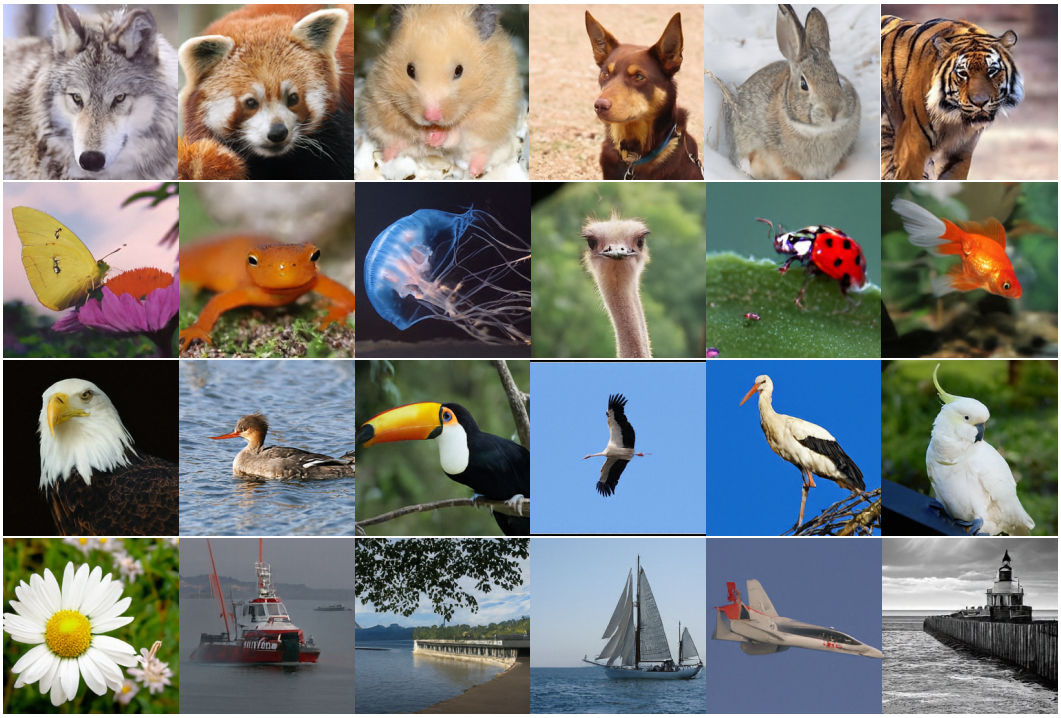


Figure 5: Qualitative Results. Selected examples on ImageNet 256×256 using CORE-L/16.

Within CRG, removing CFP increases FID to 4.27, while removing PGC increases it to 4.13. PGC alone does not improve over CDO-only, but provides an additional FID gain when combined with CFP. These results support the joint use of feature-level supervision for missing regions and correction supervision tied to pixel prediction errors within CRG.

5 Conclusion

We introduced CORE, a spatial self-supervision framework for learning contextually grounded representations in pixel-space diffusion. Our analyses show that self-distillation gains in latent space do not consistently transfer to pixel space, where high feature similarity can coexist with poor spatial correspondence. CORE combines paired generative training using CDO with CRG, which jointly supervises a shared contextual representation through cleaner-feature prediction and pixel-error-guided correction learning. Experiments on ImageNet demonstrate improved generation quality over the evaluated self-supervision baselines and better class organization along the standard generation path. CORE requires neither an external representation encoder nor an independent EMA teacher and introduces no additional inference cost. These results highlight contextual prediction and pixel-error-guided supervision as complementary tools for representation learning in pixel-space generation, linking information inferred from visible context to the pixel prediction errors.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and language polishing, improving clarity and readability. The authors reviewed and revised all AI-assisted text and take full responsibility for the accuracy and integrity of the final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 describes the model formulation, supervision targets, and training procedure. Appendix A.2 specifies the training and evaluation settings, while Section 4.2 reports component ablations. We will publicly release our source code, together with training and evaluation scripts and experiment configurations, to facilitate independent reproduction and further research.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629. IEEE, 2023.

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

Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, Yingwei Pan, Yi Peng, Zhaofan Qiu, Kai Yu, Yiheng Zhang, et al. Hidream-o1-image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

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

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv:2202.03026*, 2022.

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*, pp. 36136–36146, June 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. 2021.

Haiwen Diao, Jiahao Wang, Chenjing Ding, Hanming Deng, Jiangnan Chen, Ruixi Zhang, Ruohui Wang, Wenwen Tong, Xiangyu Fan, Yubo Wang, et al. Sensenova-u1.5: Towards native unified visual intelligence. *arXiv preprint arXiv:2609.11929*, 2026a.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026b.

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 International Conference on Machine Learning*, 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*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020.

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

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler Diffusion (SiD2): 1.5 FID on ImageNet512 with pixel-space diffusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2025.

Dengyang Jiang, Ruoyi Du, Zhennan Chen, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Xiangpeng Yang, Huanqia Cai, Aiming Hao, Yuming Jiang, et al. An empirical study of training pixel-space text-to-image diffusion models. *arXiv preprint arXiv:2608.16887*, 2026a.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Dengyang Jiang, Mengmeng Wang, Harry Yang, and Jingdong Wang. From sra to self-flow: Data augmentation or self-supervision? *arXiv preprint arXiv:2607.02508*, 2026c.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

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, 2025.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. 2024.

jiachen lei, Keli Liu, Julius Berner, Y HoiM, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no vae: End-to-end pixel-space generative modeling via self-supervised pre-training. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, 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 with latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272. IEEE, 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*, pp. 36115–36125, June 2026.

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

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *Advances in Neural Information Processing Systems*, volume 37, pp. 125441–125468, 2024.

Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, et al. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant Transformers. In *European Conference on Computer Vision*, 2024.

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*, pp. 43600–43610, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*, 2021.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

William Peebles and Saining Xie. Scalable diffusion models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 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 conference on computer vision and pattern recognition*, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv:2510.15301*, 2025.

Inkyu Shin, Chenglin Yang, and Liang-Chieh Chen. Deeply supervised flow-based generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16535–16544. IEEE, 2025.

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

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*, 2026a.

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

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.

Changyao Tian, Chenxin Tao, Jifeng Dai, Hao Li, Ziheng Li, Lewei Lu, Xiaogang Wang, Hongsheng Li, Gao Huang, and Xizhou Zhu. Addp: Learning general representations for image recognition and generation with alternating denoising diffusion process. *arXiv:2306.05423*, 2023.

Hainuo Wang, Mingjia Li, and Xiaojie Guo. Wit: Waypoint diffusion transformers for alleviating trajectory conflict in pixel-space image generation. *arXiv preprint arXiv:2603.15132*, 2026a.

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Sra 2: Variational autoencoder self-representation alignment for efficient diffusion training. *arXiv preprint arXiv:2601.17830*, 2026b.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, volume 2026, pp. 43559–43580, 2026c.

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*, pp. 40633–40642, 2026d.

Yuqing Wang, Zhijie Lin, Ceyuan Yang, Yang Zhao, Fei Xiao, Hao He, Qi Zhao, Zihan Ding, Fuyun Wang, Shuai Wang, et al. Representation forcing for bottleneck-free unified multimodal models. *arXiv preprint arXiv:2605.31604*, 2026e.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14668–14678, 2022.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 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*, 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*, pp. 14273–14282, June 2026.

Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, et al. What matters for diffusion-friendly latent manifold? prior-aligned autoencoders for latent diffusion. *arXiv preprint arXiv:2605.07915*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Guanghao Zheng, Yuchen Liu, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Mcdit: Contextual enhancement via clean-to-clean reconstruction for masked diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 97353–97377, 2024.

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

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv:2111.07832*, 2021.

Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. Sdit: Unleashing the power of self-supervised discrimination in diffusion transformer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8435–8445. IEEE, 2024.

A Appendix

A.1 Training Algorithm

Algorithm 1 summarizes a training step of CORE, combining paired generative training with CDO and contextual representation grounding through CRG.

Algorithm 1 A training step of CORE

Require: Clean-image minibatch $\mathcal{B} = \{(x_0, c)\}$; shared generator f_θ; spatial prediction parameters ϕ, including mask token m.
Require: Mask ratio ρ, prefix depth K, cleaner feature layer ℓ_c, and weights $\eta, \lambda_{\mathrm{CFP}}, \lambda_{\mathrm{PGC}}$.
Ensure: Updated parameters θ and ϕ.

CDO and paired generative training

Apply class dropout once and reuse the resulting c in all paths.
Sample $t \sim p(t)$ and independent $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, \sigma^2 I)$ for each image.
$x_t \leftarrow t x_0 + (1-t)\epsilon_1$; $y_b \leftarrow f_\theta(x_t, t, c)$.
$\epsilon^+ \leftarrow (\epsilon_1 + \epsilon_2)/\sqrt{2}$; $t^+ \leftarrow t/\big(t + (1-t)/\sqrt{2}\big)$.
$x_t^+ \leftarrow t^+ x_0 + (1-t^+)\epsilon^+$.
$y_c \leftarrow f_\theta(x_t^+, t^+, c)$, retaining image features H^c after layer ℓ_c.
$\mathcal{L}_{\mathrm{gen}} \leftarrow \langle \ell_{\mathrm{gen}}(y_b, x_0; t) + \eta\, \ell_{\mathrm{gen}}(y_c, x_0; t^+) \rangle_{\mathcal{B}}$.

Contextual representation grounding (CRG)

Sample exactly ρN masked positions $\mathcal{M}$ independently for each image.
$\widetilde{e} \leftarrow \mathrm{MaskTokens}(E_\theta(x_t), \mathcal{M}, m) + p$. ▷ Replace embeddings, then add positions
$H^m \leftarrow f_\theta^{1:K}(\widetilde{e}, t, c; A)$, using the attention mask in Eq. 5.
$Z_i \leftarrow \mathrm{GELU}(W_p\, \mathrm{LN}(H_i^m) + b_p)$.
$\widehat{h}_i \leftarrow W_h Z_i + b_h$; $\widehat{q}_i \leftarrow W_q Z_i + b_q$.

Contextual feature prediction (CFP)

$T_i \leftarrow \mathrm{sg}(\mathrm{LN}(H_i^c))$.
$\mathcal{L}_{\mathrm{CFP}} \leftarrow \left\langle 1 - \left\langle \widehat{\widehat{h}_i}, \overline{T_i} \right\rangle \right\rangle_{\mathcal{M}}$.

Pixel-guided correction (PGC)

$r_i \leftarrow \mathrm{sg}(\mathcal{P}_i(x_0 - y_b))$; $d_i \leftarrow \mathrm{sg}(\mathcal{P}_i(y_c - y_b))$ for each patch i.
$\alpha_i \leftarrow \mathrm{clip}_{[0,1]}\big(\langle r_i, d_i \rangle / (\|d_i\|_2^2 + \varepsilon_q)\big)$; $q_i \leftarrow \mathrm{sg}(\alpha_i d_i)$.
$R_{\mathcal{M}} \leftarrow \max(\mathrm{sg}[\mathrm{MSE}_{\mathcal{M}}(r, 0)], \varepsilon_r)$.
$\mathcal{L}_{\mathrm{PGC}} \leftarrow \mathrm{MSE}_{\mathcal{M}}(\widehat{q}, q)/R_{\mathcal{M}}$.

Joint optimization

$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{gen}} + \lambda_{\mathrm{CFP}}\mathcal{L}_{\mathrm{CFP}} + \lambda_{\mathrm{PGC}}\mathcal{L}_{\mathrm{PGC}}$.
Update θ and ϕ using $\nabla_{\theta,\phi}\mathcal{L}$.

Inference uses only the original generator f_θ, without masking or spatial prediction heads.

A.2 Experimental Setup

We evaluate CORE on ImageNet 2012 (Russakovsky et al., 2015) using JiT (Li & He, 2026) as the pixel-space backbone. Our main comparisons use JiT-B/16 at 256×256 resolution, trained for 200 and 600 epochs. We compare with JiT, Self-Flow (Chefer et al., 2026), SRA (Jiang et al., 2026b), DTS+AS, and the complete SRA+DTS+AS method (Jiang et al., 2026c). We further evaluate JiT-L/16 at 256×256 and JiT-B/32 at 512×512, comparing CORE with JiT and SRA+DTS+AS to examine scaling in model size and image resolution. At the L/16 scale, we additionally compare with PixelREPA (Shin et al., 2026) at 600 epochs, which uses external pretrained representations.

Our training follows the JiT recipe with AdamW (Loshchilov & Hutter, 2019), a global batch size of 1024, a learning rate of 2×10^{-4}, and zero weight decay. The learning rate remains constant after a five-epoch warmup, and class conditioning is dropped with probability 0.1. CORE uses the JiT-B/16 settings in Section 3.3. For JiT-L/16, we use $K = 8$ shared early blocks and extract cleaner features after block $\ell_c = 16$. JiT-B/32 retains $K = 4$ and $\ell_c = 8$, with a noise scale of 2.0 for 512×512 images. Full CORE models at all scales use a 128-dimensional bottleneck, a 25% masking ratio, and auxiliary loss weights $\lambda_{\mathrm{CFP}} = \lambda_{\mathrm{PGC}} = 0.05$. We use projection dropout of 0.2 for CORE-L/16 and 0 for CORE-B/16 and CORE-B/32.

We measure generation quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) on $50{,}000$ class-balanced samples. Our evaluations use the standard JiT inference path with a 50-step Heun sampler and classifier-free guidance over the interval $[0.1, 1.0]$. For each method, we report the lowest FID among the EMA and CFG configurations evaluated in the corresponding comparison, together with IS at the same operating point. EMA1 and EMA2 have decay rates 0.9999 and 0.9996, respectively. Reference methods follow their respective evaluation protocols. All CORE-specific prediction heads are disabled during sampling. Table 4 summarizes the architecture, training, grounding, and sampling configurations of CORE.

Table 4: Experimental configurations of CORE on ImageNet.

	CORE-B/16	CORE-L/16	CORE-B/32
Architecture			
Transformer depth	12	24	12
Hidden dimension	768	1024	768
Attention heads	12	16	12
Image resolution	256×256	256×256	512×512
Patch size	16×16	16×16	32×32
Image-token grid		16×16	
Patch-embedding bottleneck		128	
Projection dropout	0.0	0.2	0.0
Attention dropout		0.0	
Training			
Reported training epochs		200, 600	
Warmup epochs		5	
Optimizer		AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.95$	
Global batch size		1024	
Learning rate		2×10^{-4}	
Learning rate schedule		Constant after warmup	
Weight decay		0	
EMA decay rates		$\{0.9999, 0.9996\}$	
Timestep distribution		$\mathrm{logit}(t) \sim \mathcal{N}(-0.8, 0.8^2)$	
Noise scale σ	1.0	1.0	2.0
Denominator floor τ		0.05	
Class dropout probability		0.1	
Cleaner-view loss weight η		0.25	
Random seed		0	
Contextual Representation Grounding			
Mask ratio ρ		0.25	
Shared prefix depth K	4	8	4
Cleaner-feature layer ℓ_c	8	16	8
Shared bottleneck dimension		128	
CFP loss weight λ_{CFP}		0.05	
PGC loss weight λ_{PGC}		0.05	
Correction readout initialization		Zero weights and biases	
Sampling and Evaluation			
ODE solver		Heun	
Sampling steps		50	
Timestep grid		Uniform in $[0, 1]$	
CFG interval (Kynkäänniemi et al., 2024)		$[0.1, 1.0]$	
EMA and CFG selection		Minimum FID over evaluated settings	
Evaluation samples		$50{,}000$, class-balanced	
Auxiliary prediction heads		Disabled during sampling	

Table 5: Parameter and inference-cost comparison for JiT-B/16. Added parameters are measured relative to JiT. EMA teacher indicates its use for representation supervision. Inference Δ denotes the change in inference FLOPs relative to JiT under identical sampling settings.

Method	Trainable (M)	Added (M)	EMA teacher	Inference Δ
JiT (Li & He, 2026)	131.12	0.000	No	0%
Self-Flow (Chefer et al., 2026)	133.49	2.362	Yes	0%
SRA (Jiang et al., 2026b)	133.49	2.362	Yes	0%
DTS+AS (Jiang et al., 2026c)	133.49	2.362	Yes	0%
SRA+DTS+AS (Jiang et al., 2026c)	133.49	2.362	Yes	0%
CORE (Ours)	131.42	0.297	No	0%

A.3 Parameter and Inference Overhead

Trainable parameters Table 5 compares the parameter overhead of our reproduced methods using JiT-B/16. We count shared parameters once and include all trainable auxiliary components, while consistently excluding frozen positional embeddings and EMA copies. CORE adds only 0.297M parameters through its mask token, shared bottleneck, and two linear readouts, corresponding to 0.23% of the original generator. In comparison, each reproduced self-distillation baseline introduces a $D \to 2D \to D$ projector with 2.362M parameters. CORE therefore uses 87.4% fewer auxiliary trainable parameters and requires no independent EMA teacher for representation supervision. Both contextual objectives reuse the same early generator layers and shared bottleneck.

Inference overhead All methods in Table 5 retain the standard JiT inference path and introduce no additional inference FLOPs under identical sampling settings. In CORE, masking and contextual prediction are used only during training. Sampling uses the original generator without executing the auxiliary bottleneck or prediction heads, preserving the generator architecture and sampling procedure across all evaluated model sizes and image resolutions.

A.4 Baseline Adaptation Details

We implement the comparison methods as faithfully as possible from their original papers and available official code, retaining JiT's clean-image parameterization, velocity-space loss, and timestep distribution. Here, $t = 0$ denotes noise and $t = 1$ clean data. Student and teacher observations share the image, Gaussian noise, and class condition after label dropout. Teachers use EMA decay 0.9999 without gradients. Alignment uses image tokens and a student-side $D \to 2D \to D$ projector with SiLU activation. Layer indices are one-based. Inference retains scalar-time JiT sampling without teachers, projectors, or attention separation.

Self-Flow Self-Flow (Chefer et al., 2026) samples independent timesteps t and s from the JiT distribution. Each patch independently uses s with probability 0.25 and t otherwise, while the teacher uses the cleaner timestep $\max(t, s)$. Projected student block-4 features are aligned with teacher block-8 features using a cosine objective with weight 0.8.

SRA For SRA (Jiang et al., 2026b), the teacher timestep is $t_{\mathrm{T}} = \min(t + \delta, 1)$, where $\delta \sim \mathcal{U}(0, 0.2)$. The positive shift accounts for the reversed timestep convention relative to the original implementation. Student block-3 features are aligned with teacher block-8 features using Smooth L1 loss with transition parameter 0.05. For zero-based training epoch e, the alignment weight follows

$$\lambda_{\mathrm{SRA}}(e) = \begin{cases} 0.2, & e \le 149, \\ 0.2 \times 0.1^{(e-149)/1000}, & e > 149. \end{cases} \tag{13}$$

DTS+AS This comparison combines dual-timestep scheduling with attention separation (Jiang et al., 2026c). The 256 image tokens are randomly partitioned into groups of exactly 64 and 192 tokens, each assigned an independently sampled timestep. Student attention is restricted to tokens within the same group. Where class-conditioning tokens are introduced, each group receives its own copy using shared parameters, preventing information exchange through those tokens. The teacher

Table 6: Cleaner block-8 statistics at medium noise. Smaller D_c indicates more Gaussian-like marginals, lower two-channel tail mass indicates fewer extreme responses, and higher purity indicates stronger class organization.

Metric	SiT	JiT
Median quantile deviation D_c	0.0585	0.1641
Two-channel tail mass	1.60%	3.13%
Purity@10 (true class)	33.7%	19.1%
Purity@10 (null class)	26.8%	16.6%

retains full attention and receives a scalar-time observation at $\max(t, s)$. This variant retains cosine representation supervision with student/teacher layers $4/8$ and weight 0.8.

SRA+DTS+AS The complete configuration (Jiang et al., 2026c) uses the same exact token partition and applies attention separation to both the student and teacher. Each teacher group receives an independently sampled cleaner timestep $\min(t_g + \delta_g, 1)$, with $\delta_g \sim \mathcal{U}(0, 0.2)$, while preserving the student's group assignment. The alignment objective is cosine loss with weight 0.5, using student/teacher layers $3/8$ for B/16 and B/32 and $6/16$ for L/16.

A.5 Representation Diagnostics

We detail the representation analyses in Figure 1(b–d). The pixel-space models are JiT-B/16 and CORE-B/16 trained for 600 epochs, using the EMA checkpoint with the lowest evaluated FID for each model. The latent-space comparison uses a released pretrained SiT-B/2 checkpoint (Ma et al., 2024). All backbones remain frozen. Features are extracted through ordinary complete-input forward passes unless an input intervention is explicitly specified, without executing CORE's auxiliary prediction heads.

Noise-level matching We compare pixel and latent inputs at matched signal-to-noise ratios rather than identical timesteps. In clean-fraction coordinates, we define

$$\mathrm{logSNR}(t) = \log \frac{t^2 V}{(1-t)^2 \sigma^2}, \tag{14}$$

where V is the empirical data variance in the corresponding pixel or latent space. The calibrated variances are 0.30624 for JiT and 0.64971 for SiT, with $\sigma = 1$. Base log-SNR values are $\{-2, 0, 2\}$, denoted high, medium, and low noise, respectively. Cleaner observations increase log-SNR by $\log 2$. Thus, the cleaner log-SNR values used for distribution and class-organization analyses are approximately $\{-1.307, 0.693, 2.693\}$.

A.5.1 Feature Distribution Analysis

We extract raw cleaner features after block 8 from $1{,}024$ ImageNet validation images using their true class conditions. Each image contributes 64 preselected spatial positions and two diffusion-noise realizations, giving $131{,}072$ observations per channel. For each model and noise level, each of the 768 channels is independently standardized to zero mean and unit variance over these observations. No per-token layer normalization or ℓ_2 normalization is applied before this analysis.

We compare empirical channel quantiles with standard Gaussian quantiles. For channel c, the quantile deviation is

$$D_c = \sqrt{\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \left[\widehat{F}_c^{-1}(q) - \Phi^{-1}(q) \right]^2}, \tag{15}$$

where $\widehat{F}_c$ is the empirical distribution of the standardized channel and Φ is the standard Gaussian cumulative distribution function. The fixed grid $\mathcal{Q}$ contains 0.001, 0.005, the percentiles $0.01, 0.02, \ldots, 0.99$, and 0.995, 0.999. Smaller D_c indicates closer agreement with Gaussian channel marginals on this grid.

Table 7: Median quantile deviations across all 768 cleaner block-8 channels at different noise levels. Lower values indicate closer agreement with standard Gaussian quantiles.

	Median D_c		Median D_c^{tail}	
Noise level	SiT	JiT	SiT	JiT
High	0.0671	0.1445	0.2223	0.4627
Medium	0.0585	0.1641	0.1896	0.5147
Low	0.0449	0.1949	0.1446	0.6048

Table 8: Frozen-readout input interventions at medium noise with null-class conditioning. The same full-input readout is used for every intervention within each model. Retrieval is evaluated among 64 locations in the same image.

	Feature cosine		Retrieval (%)	
Input content	SiT	JiT	SiT	JiT
Full image	0.902	0.976	48.9	84.4
Query regions only	0.850	0.952	21.2	58.3
Context only	0.861	0.916	15.9	5.5
No image content	0.760	0.879	4.8	2.2

The Q–Q deviation plot shows $\widehat{F}_c^{-1}(q) - \Phi^{-1}(q)$ for eight channels preselected by a fixed hash rule. Zero indicates agreement with Gaussian quantiles. Shading denotes their pointwise range, not confidence intervals; the D_c ECDF covers all 768 channels. Table 7 reports median D_c and D_c^{tail}, the latter using quantiles $\{0.001, 0.005, 0.01, 0.02, 0.98, 0.99, 0.995, 0.999\}$. SiT has lower full-grid and tail deviations at all three noise levels. These statistics characterize channel marginals.

We additionally examine a fixed pair of channels, selected by the same hash rule, to characterize two-dimensional tail behavior. For each model, a ZCA whitening transform is fitted on 256 held-out images and applied to the remaining 768 images. Tail mass is the fraction of transformed features outside the circle containing 99% of the probability mass of a two-dimensional standard Gaussian, whose reference tail mass is 1%. Table 6 summarizes these statistics at medium noise, together with class neighborhood purity defined in Appendix A.5.3.

A.5.2 Input Interventions and Spatial Correspondence

Frozen linear readouts We fit an affine readout from noisy block-4 features to cleaner block-8 features at corresponding locations. We use disjoint sets of 2,048 fitting images, 512 calibration images, and 1,024 test images, with 64 preselected query positions per image. Fitting uses one noise realization per image, while calibration and testing use two. Inputs and targets first receive token-wise layer normalization without learned affine parameters and are then standardized channel-wise using fitting-set statistics. Ridge regression uses an unregularized intercept, with regularization selected from $\{10^{-4}, 10^{-2}, 1\}$ by calibration MSE. Predictions are mapped back to the layer-normalized target coordinates before evaluation. A separate readout is fitted for each model, noise level, and class-conditioning setting.

Input interventions Readouts fitted on complete inputs remain fixed across four interventions: full image, query regions only, context only, and no image content. Removed RGB regions are replaced with mid-gray before corruption and, for SiT, before VAE encoding. Noisy and cleaner views use independent diffusion noise. Student interventions share diffusion noise and SiT posterior sampling noise, while cleaner targets remain fixed and are extracted from intact images.

Similarity and retrieval Feature similarity is the mean cosine similarity between readout predictions and their corresponding cleaner targets. Spatial retrieval instead compares each prediction with all 64 cleaner target features from the same image and checks whether its own location has the highest cosine similarity. Exact ties receive fractional credit, giving chance performance of $1/64$. We average over query positions and noise realizations within each image before averaging across

Table 9: Class neighborhood purity at cleaner block 8. Purity is computed using ten gallery neighbors in the original 768-dimensional feature space.

Condition	Noise	Cleaner log-SNR	SiT	JiT	CORE
True class	High	-1.307	0.409	0.247	0.273
	Medium	0.693	0.337	0.191	0.218
	Low	2.693	0.319	0.165	0.197
Null class	High	-1.307	0.198	0.204	0.222
	Medium	0.693	0.268	0.166	0.190
	Low	2.693	0.272	0.144	0.164

images. Error bars in Figure 1(c) use 2,000 paired image-level bootstrap resamples to form 95% confidence intervals.

Table 8 shows that removing query regions leaves JiT's cosine similarity at 0.916, while retrieval drops from 84.4% to 5.5%. Both models retain substantial similarity even without image content, despite weak retrieval. This discrepancy motivates supervision that explicitly requires inference from visible context.

A.5.3 Class Organization

We evaluate class organization using 1,000 validation images from 20 classes, with 50 images per class. Each class is split into 25 gallery and 25 query images, yielding disjoint 500-image sets. We extract cleaner features after blocks 4 and 8. Block-8 features are used for the SiT–JiT comparison in Figure 1(b), while Figure 1(d) compares JiT and CORE using block-4 features under null-class conditioning. At each layer, we average all 256 raw image-token features and ℓ_2-normalize the resulting 768-dimensional image descriptor. Neighbors are selected by cosine similarity in this original feature space, without dimensionality reduction.

For query image j with class label c_j and its ten nearest gallery neighbors $\mathcal{N}_{10}(j)$, class neighborhood purity is

$$\text{Purity@10} = \frac{1}{10|\mathcal{I}|} \sum_{j \in \mathcal{I}} \sum_{k \in \mathcal{N}_{10}(j)} \mathbf{1}[c_j = c_k], \tag{16}$$

where $\mathcal{I}$ denotes the query-image set. Balanced random neighbors give an expected purity of 0.05. We evaluate both true-class conditioning and the trained null-class condition. In the latter setting, ground-truth labels are used only to score neighbor agreement, not as model inputs. The reported values use the fixed first noise realization for both gallery and query images. Confidence intervals use 2,000 class-stratified bootstrap resamples of query images with the gallery held fixed.

SiT serves as a latent-space reference. Relative to JiT, CORE improves class organization at the shared prefix output (Figure 1(d)) and at block 8 (Table 9). The block-8 gains hold across all three noise levels with both true-class and null-class inputs. These improvements occur along the ordinary generation path without auxiliary masking or prediction heads.

A.6 Limitation

CORE introduces additional training overhead compared with standard JiT, as paired generative training and auxiliary contextual prediction require extra forward and backward computation. This overhead is confined to training. At inference, CORE retains the original generator and sampling procedure without executing the auxiliary prediction heads, introducing no additional inference cost.

A.7 Additional Qualitative Results

Figures 6–8 present additional ImageNet samples generated by CORE. These examples illustrate variations in object appearance, fine-grained texture, and background composition.

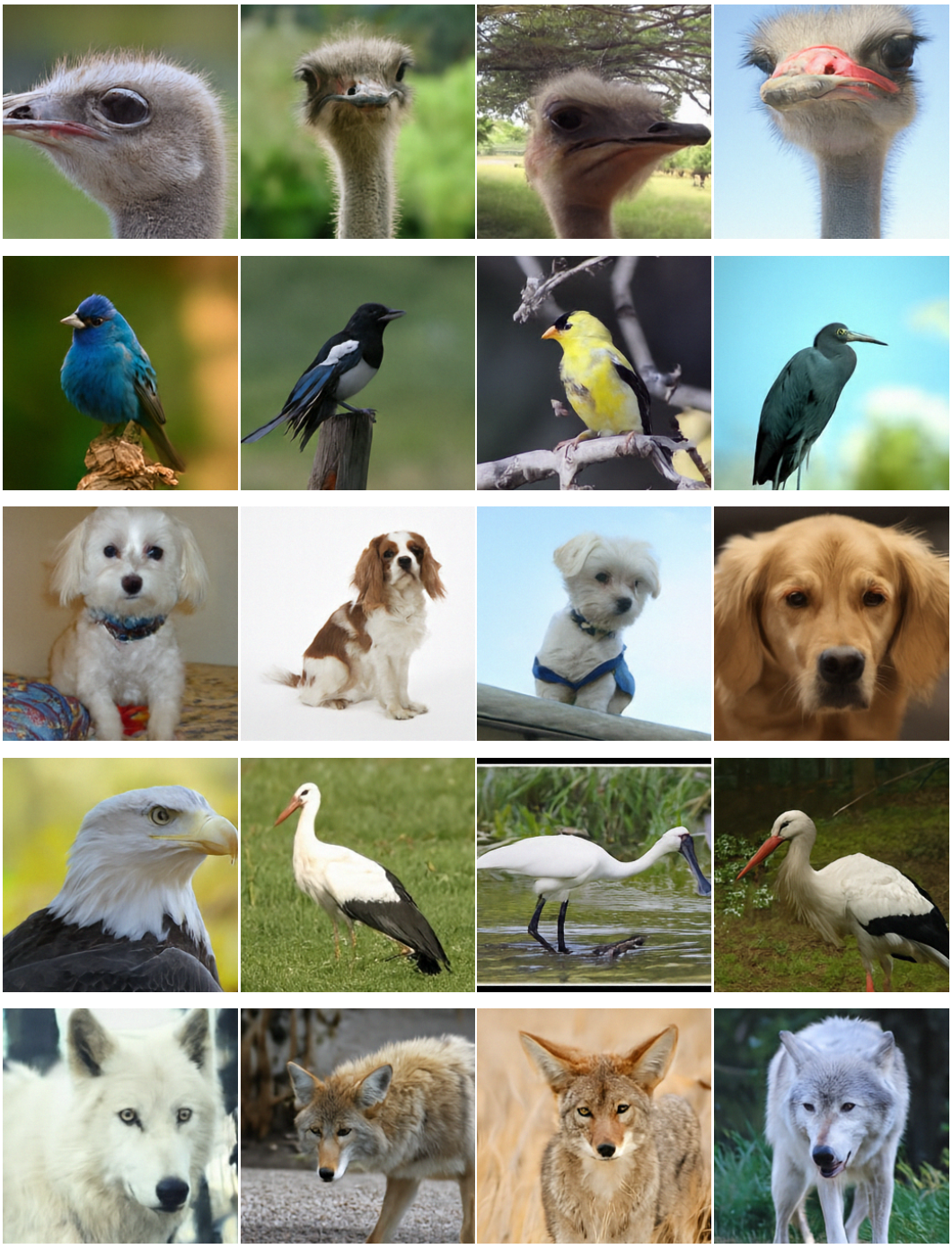


Figure 6: Additional ImageNet samples generated by CORE-L/16.

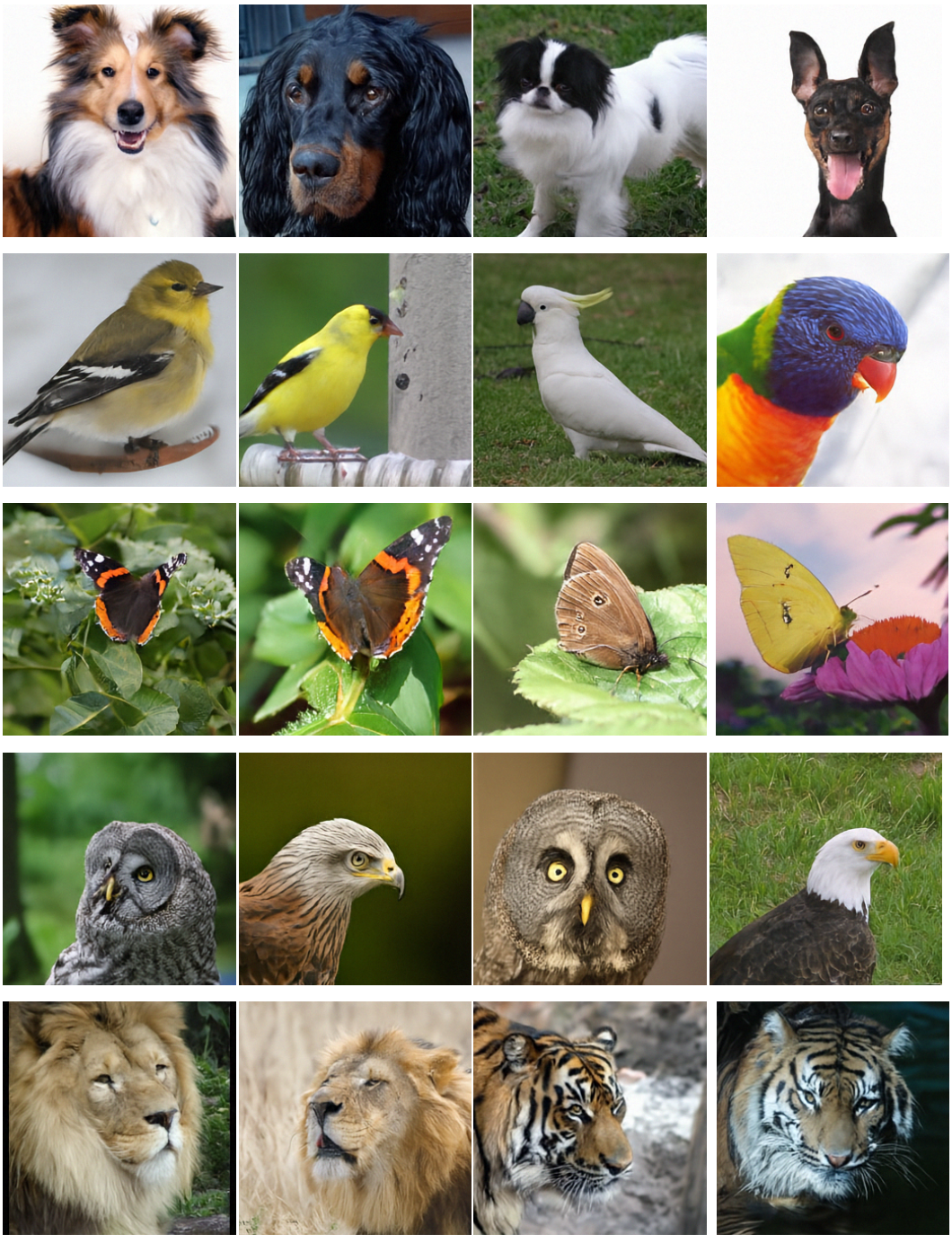


Figure 7: Additional ImageNet samples generated by CORE-L/16.

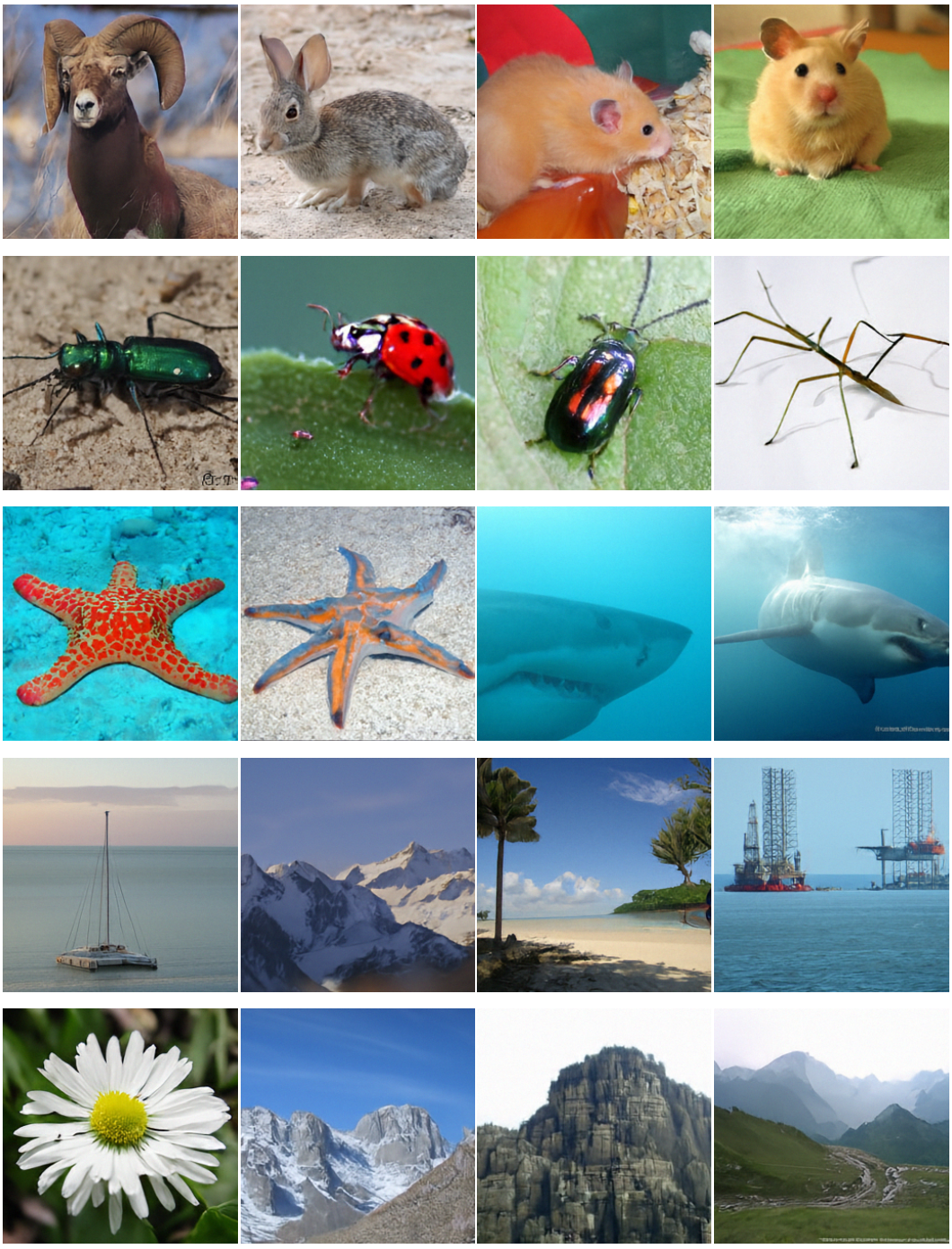


Figure 8: Additional ImageNet samples generated by CORE-L/16.