CoVA: Composing VAE Representations for Generative Learning

Anonymous authors
Paper under double-blind review



Figure 1: **High-fidelity image generation with CoVA.** Samples from SiT-XL/2 with CoVA on ImageNet at 256×256 resolution. CoVA produces images with fine-grained details, coherent structures, and strong semantic consistency.

Abstract

Representation alignment with pretrained visual encoders improves diffusion transformer training but requires auxiliary encoders and extra training overhead. Recent approaches instead exploit representations from the generative model's own variational autoencoder (VAE). However, reconstruction-oriented VAE features are not optimized for the semantic discrimination and spatial structure needed for generative learning. We introduce **CoVA** (**Co**mposed **VA**E Representations), a lightweight framework that composes VAE features online into stage-wise guidance for flow-matching training. A small latent composer is jointly trained with class supervision, masked latent reconstruction, and horizontal-flip consistency, yielding semantically discriminative, spatially structured dense features and sample-dependent class posteriors. CoVA aligns intermediate SiT features with the composed representations early in training, then removes the increasingly detrimental alignment and reuses detached composer posteriors for sample-dependent soft label conditioning. We further uncover a hidden confound in soft label conditioning: averaging class embeddings attenuates their magnitude, coupling semantic interpolation with conditioning strength. This attenuation alone can substantially improve FID while degrading Recall, revealing a non-semantic source of the apparent gain. CoVA corrects this confound by restoring the hard-label magnitude, isolating posterior-dependent directional semantics. CoVA adds only 8.5M training-only auxiliary parameters (1.3% of SiT-XL/2) and introduces no inference-time overhead. On ImageNet 256×256, CoVA reduces SiT-XL/2 FID from 17.2 to 7.4 (**57%**) in 400K steps. With classifier-free guidance, it achieves an FID of **2.03** in only 400 epochs, versus 2.06 after 1,400 epochs for vanilla SiT-XL/2.

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

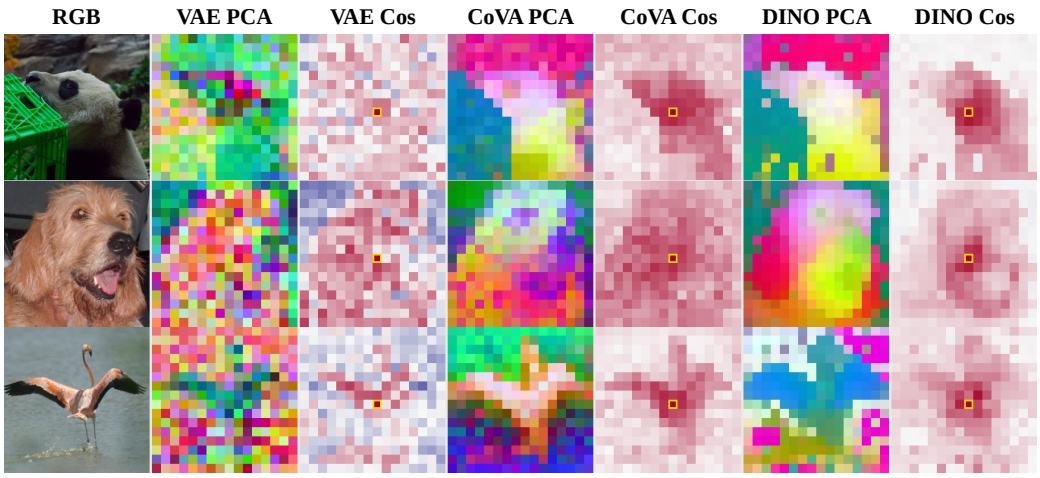


Figure 2: **CoVA reorganizes raw VAE features into more spatially coherent representations.** Each row shows one ImageNet image together with PCA visualizations and cosine-similarity maps of raw VAE latent features, CoVA-composed features, and DINOv2-B/14 features. For PCA visualization, patch features are L2-normalized and projected onto the first three principal components, which are mapped to RGB channels. Each cosine map shows the similarity between all patch tokens and the query token marked by the yellow box. Compared with raw VAE features, CoVA produces substantially more coherent and object-aligned spatial structure. Quantitatively, Spatial LDS(↑) improves from 0.28 for raw VAE features to 0.41 with CoVA, comparable to the 0.40 achieved by DINOv2.

1 Introduction

Recent advances in diffusion and flow-matching models have substantially reshaped modern image generation, delivering strong gains in visual quality, scalability, and training efficiency. From the original DDPM formulation Ho et al. (2020) and accelerated sampling methods Song et al. (2020) to flow-matching objectives Lipman et al. (2022) and scalable transformer architectures such as DiT Peebles & Xie (2023) and SiT Ma et al. (2024), these developments have steadily pushed the frontier of generative modeling. Yet recent evidence suggests that generation quality depends not only on the learning objective, but also on the structure of intermediate representations formed during optimization. Representation alignment methods such as REPA Yu et al. (2024) and its follow-up analyses Singh et al. (2026) have shown that explicitly regularizing intermediate generative features can accelerate training and improve generation quality. A common strategy is to align these features with representations extracted from pretrained visual encoders such as DINO Caron et al. (2021) or CLIP Radford et al. (2021). While effective, this approach requires an additional visual backbone throughout training, introducing nontrivial computational and memory overhead.

A natural alternative is to reuse representations available within the latent generative pipeline. Recent work has shown that VAE-derived features can provide effective intermediate supervision without relying on an external visual encoder Leng et al. (2025); Wang et al. (2026a). However, VAE representations are learned primarily for reconstruction rather than semantic organization Zheng et al. (2026), and are therefore not necessarily optimal supervision targets for a generative transformer. This motivates a fundamental question: *can reconstruction-oriented VAE features be lightly reorganized into representations that are better suited for generative learning?* To address this, we introduce **CoVA** (**Co**mposed **VA**E Representations), a lightweight framework that learns to compose VAE representations online during flow-matching training. CoVA augments the VAE features with a latent composer trained using complementary semantic and structural objectives, including class supervision, masked latent reconstruction He et al. (2022), and horizontal-flip consistency. As illustrated in Fig. 2, this lightweight composition transforms the weakly organized spatial structure of raw VAE latents into coherent, object-aligned representations. Motivated by the finding of iREPA Singh et al. (2026) that spatial structure is critical to effective representation alignment, we use these composed dense features as supervisory targets for an intermediate SiT layer through a lightweight projector. The

composer is trained online alongside the generative model using signals available in the training pipeline, avoiding dependence on a pretrained DINO-style or CLIP-style teacher.

Our experiments further reveal that even improved representation supervision is not uniformly beneficial throughout optimization. Composed VAE features provide clear gains during the earlier stages of training, whereas prolonged alignment eventually degrades generation performance, consistent with recent observations Wang et al. (2026b) on REPA. This suggests that representation guidance is better viewed as a stage-dependent signal than as a persistent constraint. It also raises a natural follow-up question: *once explicit alignment ceases to help, can the semantic information learned by the composer still benefit the generator without constraining its internal feature space?* CoVA addresses this by shifting the role of the composer from representation supervision to conditioning. At later stages, CoVA removes the intermediate alignment objective and reuses the composer's detached class posterior for sample-dependent Soft Label Conditioning. The VAE-derived signal thus evolves from shaping internal features to modulating class conditioning as training progresses.

A closer analysis reveals a hidden confound in naive soft label conditioning. Forming a soft condition by averaging class embeddings changes two factors simultaneously: its semantic direction and its magnitude. The latter unintentionally alters the strength of class conditioning, making it difficult to distinguish gains from semantic interpolation from those caused by weaker or stronger conditioning. Empirically, naive soft conditioning lowers FID Heusel et al. (2017) and improves Precision, but also reduces Recall Kynkäänniemi et al. (2019), revealing a clear quality–diversity trade-off. Controlled ablations show that this trade-off is largely driven by the unintended magnitude change. CoVA therefore explicitly restores the condition magnitude, removing this confounding effect and allowing the posterior to contribute primarily through its sample-dependent semantic direction.

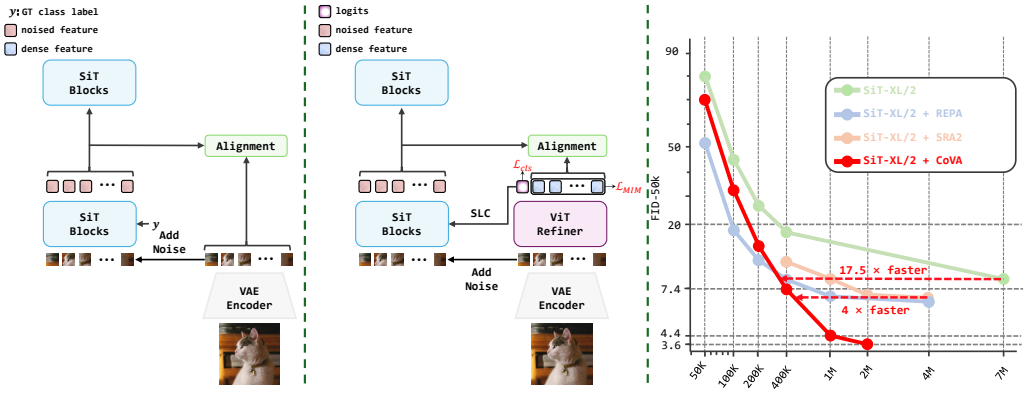
CoVA introduces only 8.5M (1.3% of SiT-XL/2) training-only auxiliary parameters, including a 5.6M latent composer and a 2.9M alignment projector, incurs only 7.2% additional training-time GFLOPs, and is discarded at inference. Across SiT model scales, CoVA consistently improves generation quality under matched training budgets. On ImageNet, it reduces SiT-XL/2 FID from 17.2 to 7.4 at 400K steps, while reaching 2.03 with classifier-free guidance in only 400 epochs. These results demonstrate that VAE representations, when appropriately composed and reused across training stages, can provide effective generative supervision without auxiliary visual encoders.

In summary, our contributions are as follows:

- We introduce a lightweight online latent composer that reorganizes reconstruction-oriented VAE latents into semantically discriminative and spatially structured dense representations, providing effective intermediate supervision without an external visual encoder.
- Building on the stage-dependent nature of representation alignment, we show that VAE-derived information can play complementary roles across optimization. CoVA uses composed dense features for early representation shaping, then removes the feature constraint and later reuses the same composer through its detached, sample-dependent class posterior.
- We identify a hidden magnitude confound in soft label conditioning: posterior averaging changes not only semantic direction but also condition magnitude, creating an unintended conditioning-strength shortcut. By restoring the hard-label magnitude, CoVA isolates the posterior-dependent directional signal and substantially mitigates the resulting fidelity–coverage trade-off.

2 Related work

Diffusion Transformers and Flow-Matching Models. Diffusion models have evolved from pixel-space denoising formulations Dhariwal & Nichol (2021); Kingma & Gao; Hoogeboom et al. to latent generative models Rombach et al. (2022), substantially reducing the computational cost of high-resolution synthesis. The adoption of transformer backbones Vaswani et al. (2017); Dosovitskiy et al. (2020); Bao et al.; Hatamizadeh et al.; Gao et al. (2023); Zheng et al.; Zhu et al. has further improved scalability, leading to architectures such as DiT Peebles & Xie (2023), SiT Ma et al. (2024), PixArt-α Chen et al. (2024), Stable Diffusion 3 Esser et al. (2024), and Lumina Qin et al. (2025). In particular, DiT establishes transformer-based denoising in latent space, while SiT generalizes this design to interpolant-based and flow-based objectives. Our work builds on SiT and focuses not on modifying the generative objective or backbone, but on improving how useful representations and conditioning signals are formed during training.



(a) SRA 2 **(b) CoVA (ours)** (c) FID-50k across training steps

Figure 3: **Comparison between SRA2 and CoVA.** (a) SRA2 aligns intermediate SiT features with VAE features through a lightweight projection layer and an auxiliary feature-alignment loss. (b) CoVA first composes clean VAE latents with an online composer trained by semantic and spatial objectives, and then reuses its dense features for early representation alignment and its detached class posterior for later soft conditioning. (c) CoVA's stage-adaptive transition from representation alignment to soft conditioning enables faster training convergence and improved generation quality.

Intermediate Representation Supervision for Diffusion Training. Recent work has shown that explicitly shaping intermediate representations can substantially accelerate diffusion-transformer training. REPA Yu et al. (2024) aligns intermediate diffusion features with representations from pretrained visual encoders, while subsequent studies further investigate which properties of the target representation are most beneficial for generation Singh et al. (2026); Wu et al. (2025). In particular, iREPA highlights the importance of spatial structure in the alignment target rather than relying solely on global semantic quality Singh et al. (2026). Although effective, these methods require pretrained representation encoders throughout training. Other approaches instead derive supervision internally. SRA Jiang et al. (2025), DoD Yue et al. (2024), and SSD Ma et al. (2025a), for example, construct guidance from representations produced by the generative model or related teacher branches, reducing reliance on external foundation encoders. HASTE Wang et al. (2026b) further shows that representation alignment is strongly stage dependent: REPA provides substantial early benefits but can plateau or become detrimental when maintained later in training. CoVA shares the motivation of lightweight and stage-aware representation learning, but differs in both the source and the use of supervision: it first learns a task-adapted representation from VAE features and subsequently reuses the learned semantic information through class conditioning rather than maintaining feature alignment.

VAE-Derived Representations for Diffusion Training. Variational autoencoders are fundamental components of latent diffusion systems, providing compact image representations that can be efficiently modeled by generative networks Rombach et al. (2022); Kingma & Welling (2013). Existing studies have explored VAE-based improvements from two complementary perspectives. Latent compression methods Chen et al. (2025a;b) reduce the spatial or channel redundancy of latent tokens to improve computational efficiency, while representation-oriented methods Yao et al. (2025); Xu et al. (2025); Leng et al. (2025) aim to make the latent space more suitable for generation. For example, REPA-E Leng et al. (2025) jointly optimizes the VAE tokenizer and the latent diffusion model, whereas VAVAE Yao et al. (2025) investigates the tension between reconstruction-oriented representations and generation-oriented objectives. SRA 2 Wang et al. (2026a) demonstrates that VAE features can also provide effective visual guidance for diffusion transformers through a lightweight projection-based alignment objective, without introducing an additional large visual encoder or a second diffusion model. CoVA is most closely related to this direction, but differs in two important respects. First, rather than aligning the generator directly to raw reconstruction-oriented VAE features, CoVA learns a lightweight online composer that reorganizes them using semantic and structural objectives before they are used as supervision. Second, CoVA does not retain VAE-based feature alignment as a fixed training constraint: once alignment ceases to be beneficial, the composer's classifier posterior is reused for sample-dependent class conditioning.

3 METHOD

3.1 PRELIMINARIES

Our method is built upon Scalable Interpolant Transformers (SiT) Ma et al. (2024), which provide a common formulation for flow-based and diffusion-based generation. SiT constructs a continuous transport path between a Gaussian prior and the data distribution, and learns the corresponding vector field with a velocity-matching objective.

Let $\boldsymbol{z} \sim p_{\text{data}}$ denote a clean latent sample and let $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ be an independent Gaussian noise sample. We use the linear interpolation

$$\boldsymbol{y}_t = t\boldsymbol{z} + (1-t)\boldsymbol{\epsilon}, \qquad t \in [0,1], \tag{1}$$

which connects Gaussian noise at $\boldsymbol{y}_0 = \boldsymbol{\epsilon}$ to the clean latent at $\boldsymbol{y}_1 = \boldsymbol{z}$. Its instantaneous target velocity is $\boldsymbol{u}_t = \boldsymbol{z} - \boldsymbol{\epsilon}$.

SiT parameterizes the velocity field with a neural network $\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{y}_t, t, \boldsymbol{c})$, where $\boldsymbol{c}$ denotes the class condition. The base flow-matching objective is

$$\mathcal{L}_{\text{FM}}(\boldsymbol{c}) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\boldsymbol{z},\boldsymbol{\epsilon}}\left[\frac{1}{n}\left\|\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{y}_t, t, \boldsymbol{c}) - (\boldsymbol{z}-\boldsymbol{\epsilon})\right\|_2^2\right], \tag{2}$$

where n is the number of latent elements. Generation proceeds from noise toward the data endpoint using the learned field. Our method retains the original SiT formulation and instead changes how VAE-derived information is used during training: first as composed intermediate supervision, and later as posterior-based class conditioning.

3.2 COMPOSING VAE REPRESENTATIONS

Directly using VAE features as representation targets is efficient, but these features are learned primarily for reconstruction rather than semantic organization. CoVA therefore learns a lightweight online composer that reorganizes the clean VAE latent into a representation better suited for intermediate supervision.

Let $\boldsymbol{Z} \in \mathbb{R}^{N\times d_z}$ denote the patchified clean VAE latent, where N is the number of spatial latent patches and d_z is the patch dimension. A lightweight ViT-based composer Dosovitskiy et al. (2020) F_θ produces dense patch features $\boldsymbol{H} \in \mathbb{R}^{N\times d_h}$ and class logits $\boldsymbol{o} \in \mathbb{R}^K$:

$$(\boldsymbol{H}, \boldsymbol{o}) = F_\theta(\boldsymbol{Z}), \qquad \boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N], \tag{3}$$

where K denotes the number of classes.

The composer is trained online with complementary semantic and spatial objectives:

$$\mathcal{L}_{\text{comp}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{mim}}\mathcal{L}_{\text{mim}} + \lambda_{\text{flip}}\mathcal{L}_{\text{flip}}. \tag{4}$$

Here, $\mathcal{L}_{\text{cls}}$ is a label-smoothed classification loss Szegedy et al. (2016), $\mathcal{L}_{\text{mim}}$ performs masked latent reconstruction following masked autoencoding He et al. (2022), and $\mathcal{L}_{\text{flip}}$ encourages spatial consistency under horizontal flipping. These objectives respectively promote class-level semantics, local latent structure, and spatial equivariance in the dense representation. Detailed formulations and implementation settings are provided in the appendix.

We use the composed dense features to supervise an intermediate SiT layer following the representation-alignment formulation of REPA Yu et al. (2024). Let $\boldsymbol{G}^{(\ell)} = [\boldsymbol{g}_1^{(\ell)}, \ldots, \boldsymbol{g}_N^{(\ell)}]$ denote the N patch-token features produced by the ℓ-th SiT block. Following REPA, a lightweight MLP projector Π maps the SiT features into the composer feature space, and spatially corresponding tokens are aligned using cosine distance:

$$\mathcal{L}_{\text{align}} = \frac{1}{N}\sum_{i=1}^{N} d_{\cos}\left(\Pi(\boldsymbol{g}_i^{(\ell)}), \text{sg}(\boldsymbol{h}_i)\right), \tag{5}$$

where $d_{\cos}(\cdot,\cdot)$ denotes cosine distance and $\text{sg}(\cdot)$ is the stop-gradient operator. Thus, $\mathcal{L}_{\text{align}}$ updates SiT and the projector, but does not back-propagate into the composer; the composer itself is optimized only by $\mathcal{L}_{\text{comp}}$.

3.3 Stage-Adaptive Supervision and Soft Label Conditioning

Representation alignment is effective early in training but becomes detrimental when maintained for too long. CoVA therefore disables alignment after a transition step s_τ and reuses the composer's semantic prediction for sample-dependent class conditioning.

For a training sample j, the composer produces the detached posterior

$$\boldsymbol{q}_j = \mathrm{sg}\left(\mathrm{softmax}(\boldsymbol{o}_j)\right), \qquad \boldsymbol{q}_j \in \mathbb{R}^K. \tag{6}$$

Unlike fixed label smoothing, $\boldsymbol{q}_j$ is sample-dependent and captures the composer's predicted relations among classes.

Let $\boldsymbol{E} \in \mathbb{R}^{K \times d_c}$ denote the SiT class-embedding table. With probability p_s, the hard-label condition is replaced by the naive soft condition

$$\bar{\boldsymbol{c}}_j = \boldsymbol{q}_j^\top \boldsymbol{E}. \tag{7}$$

Since this interpolation changes both condition direction and magnitude, we restore its magnitude to that of the corresponding hard-label embedding:

$$\boldsymbol{c}_j^{\mathrm{soft}} = \frac{\|\boldsymbol{E}_{y_j}\|_2}{\|\bar{\boldsymbol{c}}_j\|_2 + \epsilon} \bar{\boldsymbol{c}}_j, \tag{8}$$

where y_j is the ground-truth class and ϵ ensures numerical stability. This preserves the posterior-dependent direction while removing the unintended change in conditioning strength.

During training, each sample uses either $\boldsymbol{E}_{y_j}$ or $\boldsymbol{c}_j^{\mathrm{soft}}$ according to p_s, followed by standard classifier-free label dropout. Let s_τ and s'_τ denote the alignment stopping and SLC activation steps, respectively; alignment is used for $s < s_\tau$, while sample-dependent soft conditioning is activated for $s \geq s'_\tau$.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}}(\boldsymbol{c}) + \lambda_{\mathrm{align}}(s)\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{comp}}\mathcal{L}_{\mathrm{comp}}. \tag{9}$$

Implementation details of the conditioning schedule are deferred to the appendix. At inference, the composer and alignment projector are discarded, and the model uses the standard SiT pipeline with hard-label conditioning.

4 Experiments

4.1 Setup

Experimental setup. We conduct all experiments on ImageNet Deng et al. (2009) at 256×256 resolution, following ADM preprocessing Dhariwal & Nichol (2021). Images are encoded with the pretrained Stable Diffusion VAE Rombach et al. (2022), yielding pre-extracted $4 \times 32 \times 32$ latent codes. Unless otherwise specified, we use SiT-L/2 Ma et al. (2024) with 2×2 latent patches and train for 400K iterations using SiT's linear flow-matching path and velocity parameterization. SiT and the composer are optimized with AdamW Loshchilov & Hutter (2017) at a constant learning rate of 10^{-4}, zero weight decay, and global batch size 256. We use EMA with decay 0.9999 and mixed-precision training.

The CoVA composer is a tiny ViT Dosovitskiy et al. (2020) block with width 192, 12 Transformer blocks, 3 attention heads, and MLP ratio 4.0, trained throughout the full 400K iterations. We use label smoothing 0.1, masking ratio 0.7, MSE for masked latent reconstruction, cosine distance for flip consistency, and set all auxiliary loss weights to 1.0. Following REPA Yu et al. (2024), a token-wise MLP projector maps SiT features to the 192-dimensional composer space. Features are extracted from blocks 4, 8, and 8 for SiT-B/2, SiT-L/2, and SiT-XL/2, respectively. The projector is a three-layer MLP with dimensions $D \to 1152 \to 1152 \to 192$, with SiLU activations after the first two layers, where D is the hidden dimension of the corresponding SiT backbone. Alignment is used for the first 200K iterations, after which it is disabled and the detached composer posterior is used for SLC during the last 50K steps with replacement probability 0.9 and the magnitude correction from Sec. 3.3. All methods share the same backbone, optimization schedule, training budget, and preprocessing unless stated otherwise.

Table 1: **FID comparison across training iterations for accelerated alignment methods.** All experiments are conducted on ImageNet (256×256) with a batch size of 256 and without CFG.

Method	#Params	Iter.	FID↓
SiT-B/2	130M	400K	33.0
+REPA	130M	400K	24.4
+SRA2	130M	400K	28.9
+ CoVA(ours)	130M	**400K**	**21.4**
SiT-L/2	458M	400K	18.8
+REPA	458M	400K	9.7
+SRA2	458M	200K	22.5
+SRA2	458M	400K	14.3
+ CoVA(ours)	458M	**200K**	**20.4**
+ CoVA(ours)	458M	**400K**	**9.3**
SiT-XL/2	675M	400K	17.2
+REPA	675M	400K	7.9
+REPA	675M	1M	6.4
+REPA	675M	4M	5.9
+SRA2	675M	400K	11.7
+SRA2	675M	1M	8.2
+SRA2	675M	2M	7.0
+ CoVA(ours)	675M	**400K**	**7.4**
+ CoVA(ours)	675M	**1M**	**4.4**
+ CoVA(ours)	675M	**2M**	**3.6**

Table 2: **Comparison with state-of-the-art methods on ImageNet 256×256 with CFG.** "External" indicates reliance on additional pretrained models beyond the standard generative pipeline.

Method	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑	External
Pixel diffusion							
ADM-U	400	3.94	6.14	186.7	0.82	0.52	✗
VDM++	560	2.40	–	225.3	–	–	✗
Simple diffusion	800	2.77	–	211.8	–	–	✗
CDM	2160	4.88	–	158.7	–	–	✗
PixelFlow-XL/4	320	1.98	–	282.1	0.81	0.60	✗
DiP-XL/16	320	1.98	–	282.9	0.80	0.62	✗
PixNerd-XL/16	320	1.95	–	298.0	0.80	0.60	✓
PixelGen-XL/16	160	1.83	–	293.6	0.79	0.63	✓
JiT-G/16	600	1.82	–	292.6	–	–	✗
DeCo-XL/16	800	1.62	4.41	301.0	0.80	0.62	✓
Latent diffusion, Transformer + U-Net hybrid							
U-ViT-H/2	240	2.29	5.68	263.9	0.82	0.57	✗
DiffiT	–	1.73	–	276.5	0.80	0.62	✗
MDTv2-XL/2	1080	1.58	4.52	314.7	0.79	0.65	✗
Latent diffusion, Transformer							
MaskDiT	1600	2.28	5.67	276.6	0.80	0.61	✗
SD-DiT	480	3.23	–	–	–	–	✗
DiT-XL/2	1400	2.27	4.60	278.2	0.83	0.57	✗
SiT-XL/2	1400	2.06	4.50	270.3	0.82	0.59	✗
+ REPA	800	1.42	4.70	305.7	0.80	0.65	✓
+ REG	800	1.36	4.25	299.4	0.77	0.66	✓
+ SRA2	200	1.98	4.53	284.5	0.83	0.59	✗
+ SRA2	400	1.84	4.50	306.5	0.82	0.61	✗
+ CoVA (ours)	200	2.19	4.49	287.9	0.80	0.60	✗
+ CoVA (ours)	400	2.03	4.52	316.4	0.80	0.62	✗

Generation quality assessment. We report FID Heusel et al. (2017), sFID Nash et al. (2021), IS Salimans et al. (2016), Precision, and Recall Kynkäänniemi et al. (2019). We generate 50K samples from the EMA model using the SDE Euler–Maruyama solver with 250 integration steps, and evaluate against ImageNet reference statistics. Classifier-free guidance Ho & Salimans (2022) is disabled unless otherwise specified, and the same protocol is used for all methods.

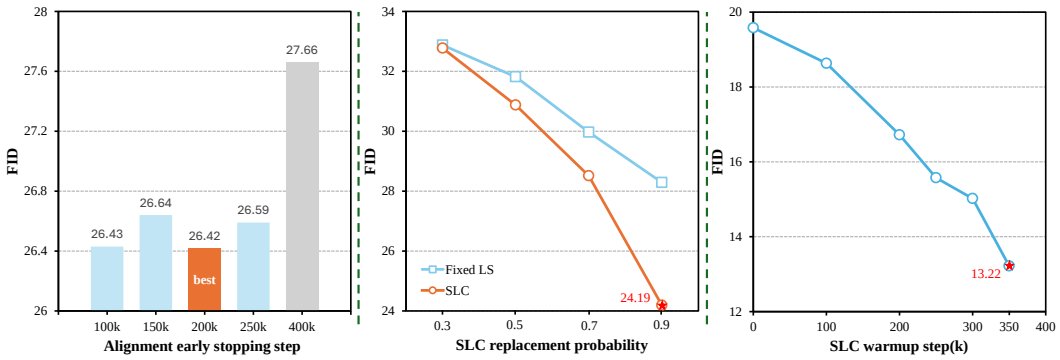
4.2 System-Level Comparisons

Comparison with SOTA methods. Tables 1 and 2 compare CoVA with representative generative models Chen et al. (2025c;d); Wang et al. (2025); Ma et al. (2026); Li & He (2025); Ma et al. (2025b) and representation-alignment methods on ImageNet 256 × 256, without and with classifier-free guidance, respectively. Under the same 400K-step budget, CoVA consistently improves FID across SiT-B/2, SiT-L/2, and SiT-XL/2, from 33.0 to 21.4, 18.8 to 9.3, and 17.2 to 7.4, respectively. With CFG, CoVA achieves an FID of 2.03 and an IS of 316.4 in only 400 epochs, compared with 2.06 and 270.3 for vanilla SiT-XL/2 after 1,400 epochs. These results suggest that CoVA's largest advantage appears in improving the underlying unguided generative model, while strong classifier-free guidance partially narrows the differences among training methods.

Training computational cost. Training-time computational overhead. CoVA avoids the external pretrained representation encoder used by methods such as REPA and instead introduces a lightweight online latent composer. As shown in Table 3, CoVA requires no external model and introduces only 8.5M trainable auxiliary parameters, substantially fewer than the 94M additional parameters required by REPA and the 18M auxiliary parameters used by SRA2. On SiT-XL/2, this increases the training computation from 114.46 to 122.69 GFLOPs (+7.2%), reduces training throughput from 8.1 to 7.0 iter/s (−14%), and increases latency from 18.05 to 21.17 ms (+17%). This overhead remains substantially lower than that of REPA, which relies on a large external encoder. Importantly, all CoVA-specific modules are used only during training and are discarded at inference, leaving the original SiT sampling pipeline unchanged.

Table 3: **Training-time computational overhead on SiT-XL/2.** EFP denotes additional feature-processing parameters, reported as *external model parameters* + *trainable auxiliary parameters*.

Method	#EFP (M)	TS (Iter/s)↑	GFLOPs↓	Latency (ms)↓
SiT-XL/2	0+0	8.1	114.46	18.05
+REPA	86+8	6.3 (-22%)	138.50 (+21%)	22.81 (+26%)
+SRA2	0+18	7.2 (**-11%**)	118.55 (**+4%**)	19.20 (**+6%**)
+CoVA	0+8.5	7.0 (**-14%**)	122.69 (**+7.2%**)	21.17 (**+17%**)



(a) Effect of alignment early stopping **(b) SLC versus Fixed Label Smoothing** **(c) SLC warmup Schedule**

Figure 4: **Stage-wise scheduling of representation alignment and conditioning.** (a) Early stopping consistently outperforms full-training alignment, with stable performance across a broad stopping range. (b) Sample-adaptive SLC consistently outperforms fixed label smoothing across all replacement probabilities, with the advantage becoming more pronounced as the replacement probability increases. (c) Longer naive SLC warmup consistently improves performance across the tested schedules. For both (a) and (b), the MIM mask ratio is fixed to 0.5.

4.3 Ablation Studies

All ablations use SiT-B/2 trained for 400k steps with 250 SDE function evaluations and without CFG.

Composer Objectives. We first isolate the contribution of each objective used to train the latent composer. As shown in Table 4, masked latent reconstruction alone improves FID from 33.02 to 31.92, while classification supervision alone further reduces it to 30.85. Starting from classification supervision, adding masked reconstruction or horizontal-flip consistency yields comparable improvements, and combining all three objectives achieves the best FID of 26.83, together with the highest IS and Precision. These results suggest that the objectives provide complementary supervision: classification organizes semantic information, while masked reconstruction and flip consistency preserve instance-level and spatial structure in the dense representations.

Table 4: **Ablation on composer training objectives.** All variants use the same feature-alignment configuration without SLC or early stopping. When enabled, classification uses label smoothing $\epsilon = 0.1$, and MIM uses a masking ratio of 0.7. Prec. and Rec. denote precision and recall.

$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{mim}}$	$\mathcal{L}_{\text{flip}}$	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
×	×	×	33.02	6.46	43.71	0.53	0.63
×	✓	×	31.92	6.26	48.14	0.54	**0.64**
✓	×	×	30.85	6.83	50.94	0.54	0.63
✓	✓	×	28.11	6.41	54.91	0.56	**0.64**
✓	×	✓	28.06	**6.23**	54.77	0.56	0.62
✓	✓	✓	**26.83**	6.32	**57.69**	**0.57**	**0.64**

Alignment Early Stopping. We next investigate how long the composed representation should be used as an alignment target. As shown in Fig. 4(a), early stopping consistently outperforms full-training alignment, with stable performance across a broad stopping range: stopping alignment between 100K and 250K iterations yields FIDs between 26.42 and 26.64, compared with 27.66 when alignment remains active throughout training. We therefore stop the representation-alignment objective at 200K iterations in the subsequent experiments.

SLC versus Fixed Label Smoothing. We then ask whether the benefit of soft conditioning comes from generic label smoothing or from the sample-dependent posterior produced by the online classifier. Under matched replacement probabilities, sample-adaptive SLC consistently outperforms fixed label smoothing, and the gap becomes more pronounced as the replacement probability increases, reaching 24.19 versus 28.29 FID at $p = 0.9$ in Fig. 4(b). This result shows that the gain is not explained by weakening the hard condition alone. The instance-specific posterior provides additional useful information for generation. Since sampling uses hard-label conditioning, we cap the SLC replacement probability at $p = 0.9$ to retain sufficient hard-label exposure during training and limit the train–test mismatch.

SLC Warmup Schedule. Finally, we study when the generator should start receiving the online classifier's posterior. As shown in Fig. 4(c), delaying SLC consistently improves performance across

Table 5: **Direction–magnitude decomposition of SLC.** Direction and magnitude refer to the training-time class condition. All variants use standard hard-label conditioning at inference.

Direction	Magnitude	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Hard	Hard	24.87	5.95	61.19	0.58	0.64
Soft	Soft	**12.71**	5.58	**118.34**	0.67	0.58
Hard	Soft	12.78	**5.55**	109.71	**0.68**	0.57
Soft	Hard	22.73	5.77	70.30	0.58	0.63
Shuffled soft	Hard	34.13	6.47	42.35	0.52	**0.64**

Table 6: **Sensitivity to inference-time condition magnitude.** We scale hard-label embeddings by γ at inference for a model trained without SLC, keeping all parameters fixed.

γ	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
1.00	24.87	5.95	61.19	0.58	**0.64**
1.10	18.42	5.74	80.46	0.63	0.61
1.20	13.80	5.62	102.67	0.67	0.58
1.35	**9.38**	**5.60**	**137.38**	**0.71**	0.55

the tested schedules. We attribute this trend to the stage-dependent reliability of the posterior: early in training, the composer is still evolving and its predictions may provide unstable semantic mixtures that interfere with the formation of a robust hard-label-conditioned generative trajectory. Once the composer has matured, its posterior becomes more suitable as a sample-adaptive refinement of the class condition. We therefore enable SLC only during the final 50K iterations.

4.4 Understanding Soft Label Conditioning: Disentangling Direction and Magnitude

Naive Soft Label Conditioning (SLC) averages class embeddings according to the sample-dependent composer posterior, changing both the *direction* and *magnitude* of the condition. For ground-truth class y, let $\boldsymbol{e}_h = \boldsymbol{e}_y$ and

$$\boldsymbol{e}_s = \sum_k q_k \boldsymbol{e}_k \tag{10}$$

denote the hard and naive soft conditions, respectively. We disentangle their effects by constructing

$$\boldsymbol{e}_{\text{hard-dir,soft-mag}} = \|\boldsymbol{e}_s\|_2 \frac{\boldsymbol{e}_h}{\|\boldsymbol{e}_h\|_2}, \qquad \boldsymbol{e}_{\text{soft-dir,hard-mag}} = \|\boldsymbol{e}_h\|_2 \frac{\boldsymbol{e}_s}{\|\boldsymbol{e}_s\|_2}. \tag{11}$$

Direction and magnitude play different roles. As shown in Table 5, using the soft magnitude alone reduces FID from 24.87 to 12.78, but also lowers Recall from 0.64 to 0.57. By contrast, using the soft direction with the hard-label magnitude gives FID improvement to 22.73 while preserving higher Recall. Importantly, shuffling soft directions across samples degrades FID sharply to 34.13, showing that the directional component carries sample-specific information rather than generic smoothing. Thus, the modest standalone FID gain of direction should not be interpreted as semantic irrelevance, preserving the correspondence between each sample and its posterior direction is critical. This suggests that the two components encode qualitatively different effects: magnitude mainly changes conditioning strength, whereas direction reflects posterior-dependent class structure.

Magnitude creates a conditioning-strength shortcut. Using the attenuated soft magnitude during training while reverting to the standard hard-label condition at inference introduces an implicit conditioning-strength mismatch. This alone produces a large change in the fidelity–coverage balance, even without introducing sample-dependent semantic direction. The inference-time scaling experiment in Table 6 independently confirms that the generator is highly sensitive to this coordinate: modifying the hard-label condition magnitude substantially changes FID, Precision, and Recall while keeping all model parameters fixed. Therefore, the large gain of naive SLC cannot be attributed solely to semantic interpolation.

5 Conclusion

We presented **CoVA**, a lightweight framework that reorganizes VAE representations and reuses them adaptively throughout flow-matching training. A small online composer first transforms reconstruction-oriented VAE features into more effective targets for intermediate representation alignment. As training progresses and explicit alignment becomes less beneficial, CoVA instead reuses the composer's class posterior for sample-dependent soft conditioning. We further show that naive soft label conditioning entangles semantic interpolation with changes in embedding magnitude, and introduce a simple magnitude correction to remove this confounding effect. CoVA requires no additional pretrained visual encoder and introduces no inference-time component.

AI USE STATEMENT

Generative AI tools were used for manuscript polishing, literature retrieval and discovery, and code review. All experimental implementations, results, analyses, and scientific claims were independently verified by the authors.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the CoVA architecture, training objectives, stage-adaptive training schedule, and soft label conditioning strategy. Experimental settings, including model architectures, optimization hyperparameters, alignment layers, auxiliary loss configurations, and sampling protocols, are reported in Sec. 4.1 and the Appendix.

REFERENCES

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pp. 57611–57640, 2024.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 96539–96560, 2025a.

Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025b.

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

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. *arXiv preprint arXiv:2511.18822*, 2025d.

Jia Deng, Wei Dong, Richard Socher, et al. 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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, et al. Mdtv2: Masked diffusion transformer is a strong image synthesizer. In *arXiv preprint arXiv:2303.14389*, 2023.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *ECCV*.

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*, pp. 16000–16009, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay 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*.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *NeurIPS*.

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

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, et al. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, et al. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. In *arXiv preprint arXiv:2504.10483*, 2025.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Nanye Ma, Mark Goldstein, Michael S Albergo, et al. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

Xinyin Ma, Runpeng Yu, Songhua Liu, Gongfan Fang, and Xinchao Wang. Diffusion model is effectively its own teacher. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12901–12911. IEEE, 2025a.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. *arXiv preprint arXiv:2511.19365*, 2025b.

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

Charlie Nash, Jacob Menick, Sander Dieleman, et al. Generating images with sparse representations. In *arXiv preprint arXiv:2103.03841*, 2021.

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

Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Xinyue Li, Dongyang Liu, Xiangyang Zhu, et al. Lumina-image 2.0: A unified and efficient image generative framework. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20031–20042. IEEE, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, et al. Improved techniques for training gans. In *NeurIPS*, 2016.

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*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *arXiv preprint arXiv:2010.02502*, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, 2026a.

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

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, et al. Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. *Advances in Neural Information Processing Systems*, 38:136854–136887, 2026b.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025.

Wanghan Xu, Xiaoyu Yue, Zidong Wang, Yao Teng, Wenlong Zhang, Xihui Liu, Luping Zhou, Wanli Ouyang, and Lei Bai. Exploring representation-aligned latent space for better generation. *arXiv preprint arXiv:2502.00359*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *arXiv preprint arXiv:2501.01423*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, et al. Representation alignment for generation: Training diffusion transformers is easier than you think. In *arXiv preprint arXiv:2410.06940*, 2024.

Xiaoyu Yue, Zidong Wang, Zeyu Lu, Shuyang Sun, Meng Wei, Wanli Ouyang, Lei Bai, and Luping Zhou. Diffusion models need visual priors for image generation. *arXiv preprint arXiv:2410.08531*, 2024.

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

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

Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In *CVPR*.

A EXPERIMENTAL SETUP

Hyperparameter setup. Table 7 summarizes the configurations used by CoVA across the three SiT model scales. All models operate on pre-extracted $4 \times 32 \times 32$ VAE latents and use the same 2×2 latent patchification. The composer is kept fixed across scales, while the SiT alignment layer follows the layer choices used in the main experiments. Unless otherwise specified, all models are trained for 400K iterations with AdamW Loshchilov & Hutter (2017), a learning rate of 10^{-4}, zero weight decay, and a global batch size of 256. The alignment loss is active for the first 200K iterations. SLC uses a constant replacement probability of 0.9 and is activated only during the final 50K iterations. Specifically, SLC starts at 350K iterations for the default 400K-run setting, and at $T - 50$K for longer runs with a total training budget of T iterations. SLC uses the magnitude correction described in Sec. 3.3. Unless explicitly marked as *naive SLC*, all CoVA results use the magnitude-corrected soft condition in Eq. 8. Naive SLC is used only in diagnostic analyses, including the scheduling study in Fig.4(c) and the direction–magnitude analysis in Sec. 4.4.

Table 7: **CoVA hyperparameter settings across SiT model scales.** The alignment layer is indexed by the one-based SiT Transformer block depth. The same composer and stage-wise schedule are used across scales unless otherwise specified.

Backbone	**SiT-B/2**	**SiT-L/2**	**SiT-XL/2**
Architecture			
# Params	130M	458M	675M
Input latent	$32 \times 32 \times 4$	$32 \times 32 \times 4$	$32 \times 32 \times 4$
# SiT blocks	12	24	28
Hidden dimension	768	1,024	1,152
# Attention heads	12	16	16
Latent patch size	2×2	2×2	2×2
Composer			
Feature dimension	192	192	192
# Composer blocks	12	12	12
# Composer heads	3	3	3
MLP ratio	4.0	4.0	4.0
Label smoothing	0.1	0.1	0.1
Masking ratio	0.7	0.7	0.7
λ_{mim}	1.0	1.0	1.0
λ_{flip}	1.0	1.0	1.0
Alignment and conditioning			
Alignment layer	4	8	8
Alignment weight	1.0	1.0	1.0
Alignment duration	200K	200K	200K
SLC replacement probability	0.9	0.9	0.9
SLC warmup	350K	350K	350K
Optimization and sampling			
Global batch size	256	256	256
Optimizer	AdamW	AdamW	AdamW
Learning rate	$1e{-4}$	$1e{-4}$	$1e{-4}$
Prediction	velocity	velocity	velocity
Sampler	Euler-Maruyama	Euler-Maruyama	Euler-Maruyama
Sampling steps	250	250	250

Masked latent reconstruction. To encourage the latent composer to preserve local and spatial information from the VAE representation, we adopt a masked latent reconstruction objective. We randomly mask a subset of latent tokens, denoted by $\mathcal{M}$, and use the composer to infer their representations from the remaining visible tokens. A lightweight prediction head $D(\cdot)$ maps the composed feature h_i at each masked position back to the VAE latent space. The reconstruction loss is defined as

$$\mathcal{L}_{\mathrm{mim}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \|D(h_i) - Z_i\|_2^2 , \tag{12}$$

where Z_i denotes the original VAE latent feature at position i. The loss is evaluated only over masked positions, encouraging the composer to capture contextual and spatial structure rather than relying solely on local appearance.

Horizontal-flip consistency. We further impose a flip-consistency objective to encourage the composed dense representation to be equivariant to horizontal flipping. Given an input latent representation and its horizontally flipped counterpart, we obtain the corresponding composer features h and h^{flip}. We then spatially flip h^{flip} back to the original coordinate system using $\mathcal{F}^{-1}(\cdot)$ and minimize their cosine distance:

$$\mathcal{L}_{\text{flip}} = \frac{1}{N} \sum_i d_{\cos} \left(h_i,\, \mathcal{F}^{-1}(h_i^{\text{flip}}) \right). \tag{13}$$

Here, N denotes the number of spatial tokens and $d_{\cos}(\cdot, \cdot)$ denotes cosine distance. This objective encourages the composer to retain spatially consistent and transformation-equivariant information in its dense representations.

B Spatial LDS Protocol

Metric and evaluation protocol. We measure the spatial organization of composer features using Spatial Local Distance Similarity (Spatial LDS) Singh et al. (2026). For each ImageNet validation image, we extract a row-major 16×16 feature grid from the deterministic mode of the pretrained VAE posterior. Each token is ℓ_2-normalized. LDS is the mean cosine similarity over unique horizontal/vertical 4-neighbor pairs (Manhattan distance 1, 480 pairs) minus the mean over directed token pairs with Manhattan distance at least 6 (53,300 pairs), self-pairs are excluded. Thus, LDS measures local-versus-distant similarity contrast rather than overall feature similarity.

We use the first 5,000 ImageNet validation images, resized and center-cropped to 256×256, identically across conditions. Composer features are the unmasked dense tokens before the classification head. DINOv2-B/14 is an external feature reference, with ImageNet normalization and resizing to 224×224 to produce the same token grid. We evaluate the 200K model weights and report per-image means. The numerical comparison is summarized in Table 8, and the corresponding qualitative comparison is shown in Figure 5.

Table 8: **Spatial LDS objective comparisons at 200K.** Near/far denote mean cosine similarity; LDS is near minus far. All values are rounded to two decimal places.

Feature / objective	Near↑	Far↓	LDS↑
Raw VAE	0.44	0.16	0.28
CE	0.48	0.19	0.29
MIM only	0.92	0.86	0.06
CE+MIM	0.60	0.24	0.36
CE+flip	0.92	0.83	0.09
CE+MIM+flip	0.66	0.25	**0.41**
DINOv2-B/14	0.64	0.25	0.40

The objective effects interact. Using CE as the anchor, adding MIM increases LDS by $+0.07$. Adding flip consistency without MIM changes LDS by -0.20. When MIM is already present, adding flip increases LDS by $+0.05$; equivalently, adding MIM to CE+flip increases LDS by $+0.31$. The difference-in-differences interaction is $+0.25$, indicating strong complementarity between MIM and flip consistency in this run-level comparison.

LDS measures local-versus-distant contrast rather than overall feature similarity. MIM-only and CE+flip have high near and far cosine similarities ($0.92/0.86$ and $0.92/0.83$, respectively), but low LDS. Their features are therefore highly similar across the grid and show limited local-versus-distant differentiation. MIM supplies patch-level reconstruction pressure, CE supplies class semantics, and flip consistency constrains how dense features transform under reflection. The results suggest that flip consistency is useful in this comparison when combined with MIM, but not in isolation.

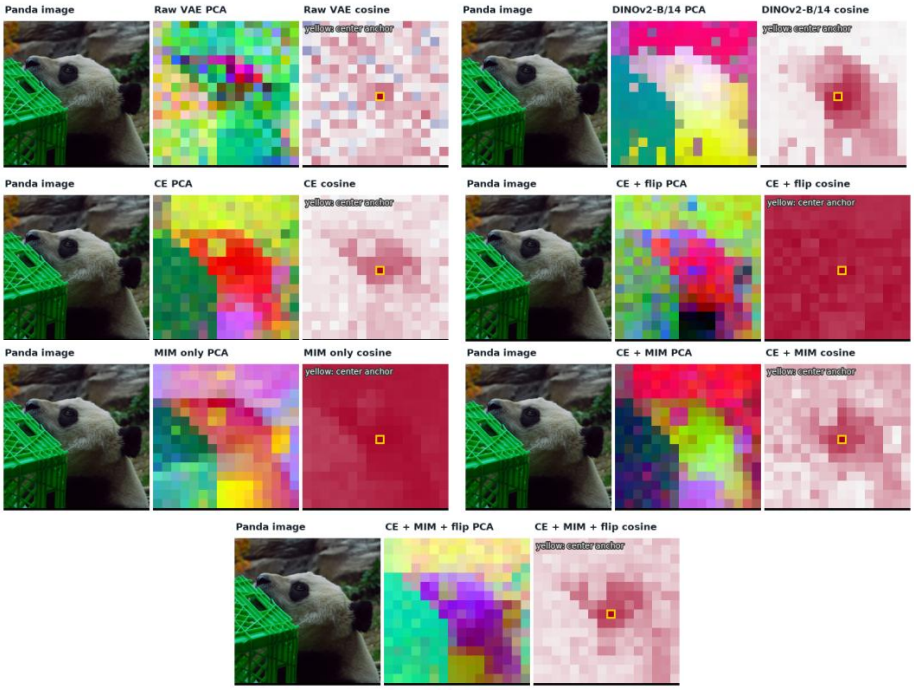


Figure 5: **PCA and cosine-map comparison for the seven table conditions.** Each panel uses the same panda image, 16×16 token grid, and center anchor.

Table 9: **Cumulative ablation of CoVA on SiT-B/2.** All models are trained for 400K iterations on ImageNet 256×256 without CFG under the same optimization and sampling protocol. Each row progressively adds one component to the preceding configuration.

Configuration	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
SiT-B/2	33.02	6.46	0.53	0.63	43.71
+ Composed representation alignment	26.83	6.32	0.57	0.64	57.69
+ Alignment early stopping	24.87	5.96	0.58	0.64	61.19
+ Soft label conditioning	21.45	5.69	0.59	0.63	74.57

C Additional Ablation Studies

All experiments in this section use SiT-B/2 trained for 400K iterations with 250 SDE sampling steps and without CFG, unless stated otherwise. The main paper presents the primary ablations on composer objectives, alignment duration, sample-adaptive versus fixed conditioning, and SLC warmup. Here, we provide additional analyses of composer capacity and robustness.

Cumulative ablation of CoVA. We progressively add the main components of CoVA to the SiT-B/2 baseline under the same 400K-step training and evaluation protocol. As shown in Table 9, composed representation alignment first reduces FID from 33.02 to 26.83. Early stopping the alignment further improves FID to 24.87. Finally, reusing the detached composer posterior for late-stage soft label conditioning yields the full CoVA model and further reduces FID to 21.45. These results show that **representation composition, stage-adaptive alignment, and late-stage conditioning provide complementary improvements and jointly contribute to the final performance of CoVA.**

Composer capacity. We first vary the composer capacity while keeping the training recipe fixed. As shown in Table 10, performance remains largely stable across a wide range of composer sizes. Increasing the capacity from the smallest to the largest configuration changes FID only marginally, from 27.52 to 27.32, while the intermediate configuration obtains 28.19. The absence of a monotonic

scaling trend suggests that **the gains of CoVA do not rely on increasing auxiliary model capacity. A compact composer already provides effective supervision.**

Table 10: **Composer capacity.** Configurations are reported as $(d_h, \text{depth}, \text{heads}, r_{\text{MLP}})$. Params, GFLOPs, and GMACs refer to the auxiliary composer.

Composer	Params	GFLOPs	GMACs	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
$(192, 12, 3, 4)$	5.59M	3.34	1.67	27.66	6.32	0.57	0.64	55.28
$(192, 8, 3, 4)$	3.81M	2.23	1.11	27.52	6.22	0.57	0.64	55.96
$(768, 3, 12, 4)$	22.26M	11.54	5.77	28.19	6.25	0.56	0.64	54.84
$(768, 4, 12, 4)$	29.35M	15.38	7.69	27.32	6.27	0.57	0.64	55.74

Table 11: **Sensitivity to composer objective hyperparameters.** We vary label smoothing η and MIM masking ratio ρ while keeping the remaining training settings fixed.

Setting	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
Label smoothing					
$\eta = 0.00$	31.91	6.78	0.54	0.64	48.64
$\eta = \mathbf{0.10}$	30.86	6.83	0.55	0.64	50.95
$\eta = 0.20$	**30.76**	6.84	0.55	0.63	50.45
$\eta = 0.30$	31.36	6.79	0.54	0.64	49.90
MIM masking ratio					
$\rho = 0.10$	31.63	6.74	0.54	0.64	49.67
$\rho = 0.30$	30.01	6.82	0.55	0.64	51.91
$\rho = 0.50$	28.56	6.53	0.56	0.64	53.96
$\rho = \mathbf{0.70}$	**28.11**	**6.41**	**0.56**	**0.64**	**54.92**

Sensitivity to composer objective hyperparameters. We further examine the sensitivity of CoVA to the main composer-objective hyperparameters. As shown in Table 11, for label smoothing, moderate values are consistently better than either no smoothing or stronger smoothing: $\eta = 0.10$ and 0.20 achieve similar performance, reducing FID from 31.91 to 30.86 and 30.76, respectively, while $\eta = 0.30$ slightly degrades performance. The MIM masking ratio has a clearer effect, with FID steadily improving from 31.63 at $\rho = 0.10$ to 28.11 at $\rho = 0.70$. Overall, CoVA is relatively insensitive to moderate changes in label smoothing, while benefiting from stronger masked reconstruction. We use $\eta = 0.10$ and $\rho = 0.70$ as the default settings.

Alignment placement and loss weights. Table 12 examines two implementation choices independently. Moving the alignment target from block 4 to deeper blocks consistently degrades performance, with FID increasing from 27.66 to 30.02 and 32.65 at blocks 6 and 8, respectively. In contrast, moderate changes to the alignment, MIM, and flip-consistency weights have a smaller effect, although none improves upon the default configuration. These results indicate that **CoVA does not require careful tuning of the auxiliary loss coefficients.**

Table 12: **Sensitivity to alignment placement and loss weights.** Each sweep varies one factor while keeping the remaining settings fixed. The default alignment block is 4, and all default loss weights are 1.0.

Setting	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
(a) Alignment block					
Block 4 (Default)	27.66	6.32	0.57	0.64	55.28
Block 6	30.02	6.43	0.55	0.64	51.09
Block 8	32.65	6.42	0.54	0.64	47.13
(b) Loss weights					
Default $(1, 1, 1)$	27.66	6.32	0.57	0.64	55.28
$\lambda_{\text{align}} = 0.5$	28.25	6.39	0.56	0.64	54.45
$\lambda_{\text{mim}} = 5$	28.73	6.39	0.56	0.64	53.75
$\lambda_{\text{flip}} = 5$	28.69	6.44	0.56	0.63	53.33

Online adaptation and SLC scheduling. Table 13 examines whether the composer should remain online and whether SLC requires more elaborate scheduling. Freezing a pre-trained composer degrades FID from 27.66 to 29.27, supporting joint online adaptation throughout training. For SLC, early or middle activation performs poorly, while delaying it to the later stage is substantially better. Conditioning the replacement probability on flow time t does not outperform the simple late-stage schedule, and Dirichlet conditioning is considerably worse. These results support a simple design: keep the composer adaptive and introduce SLC only at a late training stage, without additional t-dependent conditioning schedules.

Table 13: **Online adaptation and alternative SLC schedules.** These schedules are exploratory variants of the final late-stage SLC design.

Setting	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
Composer adaptation					
Joint online composer	27.66	6.32	0.57	0.64	55.28
Pre-train then freeze	29.27	6.40	0.56	0.63	52.30
Alternative SLC schedules					
Early activation (150K)	35.57	6.49	0.52	0.64	43.31
Middle activation	35.96	6.48	0.52	0.64	43.02
Late activation	30.57	6.14	0.55	0.63	53.25
$p \propto t$	31.97	6.38	0.54	0.63	49.80
$p \propto (1-t)$	29.23	6.01	0.56	0.62	54.88
Dirichlet, $p = 0.30$	38.80	6.40	0.51	0.63	36.61

CFG sensitivity after SLC training. We evaluate the robustness of the SLC-trained SiT-XL/2 model to inference-time CFG settings. As shown in Table 14, performance is stable over a moderate range of guidance scales. With the guidance interval fixed to $[0, 0.50]$, scales from 1.60 to 1.80 achieve FIDs between 1.99 and 2.04, indicating little sensitivity to precise CFG tuning. The lowest FID of 1.99 is obtained at scale 1.65, while scale 1.80 yields substantially higher IS and slightly higher Precision with only a minor FID difference. We therefore use $\mathrm{CFG} = 1.80$ with interval $[0, 0.50]$ for the reported results.

Table 14: **CFG sensitivity of SLC-trained SiT-XL/2.** Results are evaluated after 2M training iterations using 50K samples. The interval denotes the portion of the sampling trajectory where CFG is applied.

CFG scale	Interval	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
Guidance-scale sensitivity						
1.60	$[0, 0.50]$	2.01	4.63	0.79	**0.62**	291.74
1.65	$[0, 0.50]$	**1.99**	4.54	0.79	0.61	299.15
1.70	$[0, 0.50]$	2.01	4.56	0.79	**0.62**	304.19
1.80	$[0, 0.50]$	2.03	**4.52**	**0.80**	0.62	**316.42**
Guidance-interval sensitivity						
1.80	$[0, 0.40]$	2.12	4.61	0.78	**0.62**	278.18
1.80	$[0, 0.50]$	2.03	**4.52**	**0.80**	**0.62**	**316.42**

D Understanding the Magnitude Confound in Soft Label Conditioning

Geometry of soft class embeddings. We first analyze why posterior averaging systematically changes the magnitude of the class condition. Let $\{\boldsymbol{e}_k\}_{k=1}^{K}$ denote the learned class embeddings and let

$$\boldsymbol{e}_s = \sum_{k=1}^{K} q_k \boldsymbol{e}_k \tag{14}$$

be the naive soft condition produced from the composer posterior $\boldsymbol{q}$. For the trained SiT-XL/2 model, the class embeddings have similar magnitudes, with a mean L2 norm of 1.542 and a standard

deviation of 0.180. Meanwhile, different class embeddings are nearly orthogonal: the mean pairwise cosine similarity is only 0.016 ± 0.078.

The squared magnitude of the soft condition can be written as

$$\begin{aligned} \|\boldsymbol{e}_s\|_2^2 &= \left\| \sum_k q_k \boldsymbol{e}_k \right\|_2^2 \\ &= \sum_k q_k^2 \|\boldsymbol{e}_k\|_2^2 + \sum_{i \neq j} q_i q_j \boldsymbol{e}_i^\top \boldsymbol{e}_j . \end{aligned} \tag{15}$$

Since the class embeddings have comparable norms and small pairwise inner products, the cross terms in Eq. 15 are relatively small. Denoting the typical class-embedding norm by $\bar{r}$, we obtain the approximation

$$\|\boldsymbol{e}_s\|_2 \approx \bar{r} \sqrt{\sum_k q_k^2} = \bar{r} \|\boldsymbol{q}\|_2 . \tag{16}$$

Therefore, posterior averaging naturally couples semantic uncertainty with conditioning magnitude. A concentrated posterior approaches the magnitude of a hard-label embedding, whereas a more dispersed posterior produces a shorter conditioning vector. Consequently, naive SLC changes not only the semantic direction of the class condition, but also its effective magnitude.

Connection to classifier-free guidance. The sensitivity to condition magnitude can also be related to classifier-free guidance (CFG). Let $\boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e})$ denote the velocity prediction under condition $\boldsymbol{e}$, and let $\boldsymbol{e}_{\text{null}}$ denote the unconditional embedding. CFG produces

$$\boldsymbol{v}_{\text{CFG}} = \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}_{\text{null}}) + w \left[\boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}) - \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}_{\text{null}}) \right] , \tag{17}$$

where w is the guidance scale.

Locally linearizing the conditional prediction around $\boldsymbol{e}_{\text{null}}$ gives

$$\boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}) \approx \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}_{\text{null}}) + \boldsymbol{J}_e \left(\boldsymbol{e} - \boldsymbol{e}_{\text{null}} \right) , \tag{18}$$

where $\boldsymbol{J}_e = \partial \boldsymbol{v} / \partial \boldsymbol{e}$. Substituting this approximation into Eq. 17 yields

$$\begin{aligned} \boldsymbol{v}_{\text{CFG}} &\approx \boldsymbol{v}(\boldsymbol{x}, t, \boldsymbol{e}_{\text{null}}) + w \boldsymbol{J}_e \left(\boldsymbol{e} - \boldsymbol{e}_{\text{null}} \right) \\ &\approx \boldsymbol{v} \left(\boldsymbol{x}, t, \boldsymbol{e}_{\text{null}} + w \left(\boldsymbol{e} - \boldsymbol{e}_{\text{null}} \right) \right) . \end{aligned} \tag{19}$$

This provides a useful local interpretation of CFG as scaling the displacement of the condition from the unconditional anchor in embedding space. It does *not*, however, imply an exact equivalence between condition magnitude and CFG. The approximation is local, the soft condition is not generally collinear with the hard condition, and the limiting average class embedding does not coincide with the learned unconditional embedding. Moreover, the conditional response of a deep network is nonlinear, so $\boldsymbol{J}_e$ need not remain constant for large changes in condition magnitude.

Nevertheless, Eq. 19 suggests that condition magnitude can act as an effective conditioning-strength coordinate. This is consistent with our inference-time scaling experiment, where directly rescaling a hard-label embedding substantially changes the fidelity–coverage trade-off even though all model parameters are fixed.

Magnitude and direction play distinct roles. The analysis above explains why naive SLC contains an implicit conditioning-strength component, but does not imply that its directional variation is irrelevant. Our direction–magnitude decomposition in the main paper shows that replacing the hard-label magnitude with the attenuated soft magnitude produces a large improvement in FID, but also noticeably reduces Recall. This behavior indicates that magnitude attenuation primarily acts as a fidelity-oriented conditioning shortcut. In contrast, preserving the soft direction while restoring the hard-label magnitude yields a smaller but meaningful improvement while retaining substantially better coverage. Furthermore, shuffling soft directions across samples strongly degrades performance, demonstrating that the posterior-dependent direction contains sample-specific semantic information.

Taken together, naive SLC entangles two fundamentally different effects: *posterior-dependent semantic direction* and *posterior-dependent conditioning magnitude*. The latter can strongly alter the fidelity–coverage balance and therefore confounds the apparent gain of soft conditioning. CoVA

explicitly removes this confound by restoring the soft condition to the corresponding hard-label magnitude, while preserving its posterior-dependent direction. In this way, the soft condition contributes sample-specific semantic information without relying on unintended changes in conditioning strength.

Null-centered controls for conditioning strength. Our default magnitude correction in Eq. (8) restores the soft condition to the norm of the corresponding hard-label embedding. Since the local interpretation of classifier-free guidance is more naturally expressed relative to the learned unconditional embedding $\boldsymbol{e}_\varnothing$, we further examine whether the observed effect depends on measuring magnitude from the origin. We conduct two complementary null-centered controls, one during training and one only at inference.

Null-centered normalization during training. For the naive soft condition $\bar{\boldsymbol{c}}_j = \boldsymbol{q}_j^\top \boldsymbol{E}$, we construct a null-centered variant that explicitly preserves the distance from the learned unconditional embedding:

$$\boldsymbol{c}_j^{\text{soft,null}} = \boldsymbol{e}_\varnothing + \frac{\left\|\boldsymbol{E}_{y_j} - \boldsymbol{e}_\varnothing\right\|_2}{\left\|\bar{\boldsymbol{c}}_j - \boldsymbol{e}_\varnothing\right\|_2 + \epsilon} \left(\bar{\boldsymbol{c}}_j - \boldsymbol{e}_\varnothing\right). \tag{20}$$

By construction,

$$\left\|\boldsymbol{c}_j^{\text{soft,null}} - \boldsymbol{e}_\varnothing\right\|_2 = \left\|\boldsymbol{E}_{y_j} - \boldsymbol{e}_\varnothing\right\|_2, \tag{21}$$

so the soft condition preserves the hard-label displacement from the unconditional anchor rather than only its norm with respect to the origin.

Starting from the same 350K checkpoint, we replace the default normalization with Eq. 20 during the final 50K training iterations, while keeping all other SLC settings unchanged. As shown in Table 15, null-centered normalization achieves an FID of 21.14, closely matching the 21.45 FID of the default CoVA formulation. It also retains high Recall (0.628). This result indicates that **the improvement from corrected SLC does not depend on using the embedding-space origin as the normalization reference**; preserving conditioning strength relative to the learned unconditional embedding leads to essentially the same behavior.

Null-centered scaling at inference. We next directly perturb conditioning strength relative to the unconditional anchor while keeping the model parameters fixed. For a hard-label embedding $\boldsymbol{e}_y$, we define

$$\boldsymbol{e}_y^{(\gamma)} = \boldsymbol{e}_\varnothing + \gamma\left(\boldsymbol{e}_y - \boldsymbol{e}_\varnothing\right), \tag{22}$$

where $\gamma = 1$ recovers the standard hard-label condition. This transformation changes only the displacement from the unconditional embedding and therefore provides a direct intervention on the conditioning-strength coordinate suggested by the local CFG interpretation.

As γ increases from 1.0 to 1.35, FID decreases from 24.87 to 9.19 and Precision increases from 0.58 to 0.71, while Recall decreases from 0.64 to 0.56. The same monotonic fidelity–coverage trade-off observed with origin-centered scaling therefore remains when conditioning strength is defined relative to $\boldsymbol{e}_\varnothing$. Together with the training-time control above, these results confirm that **conditioning magnitude constitutes a genuine conditioning-strength coordinate rather than an artifact of origin-centered normalization**, while the remaining gain of magnitude-corrected SLC is attributable to the posterior-dependent directional signal.

E Supplementary Qualitative Results

We provide additional qualitative results for both image generation and spatial representation quality. Figure 8 shows randomly sampled images generated by SiT-XL/2 with CoVA, while Figure 9 extends the PCA and patch-level cosine-similarity analysis in Figure 2 to a broader set of ImageNet examples.

Table 15: **Null-centered controls for soft label conditioning.** Top: inference-time scaling of the hard-label displacement from the learned unconditional embedding using Eq. 22; all model parameters are fixed and the model is trained without SLC. Bottom: training-time SLC using the null-centered magnitude correction in Eq. 20, applied during the final 50K iterations.

Setting	FID↓	sFID↓	Prec.↑	Rec.↑	IS↑
Inference-time null-centered scaling					
$\gamma = 1.00$	24.87	5.95	0.58	0.64	61.19
$\gamma = 1.10$	18.48	5.72	0.62	0.61	81.05
$\gamma = 1.20$	13.73	5.55	0.66	0.59	104.42
$\gamma = 1.30$	10.41	5.47	0.70	0.57	129.03
$\gamma = 1.35$	**9.19**	**5.45**	**0.71**	0.56	**141.10**
Training-time null-centered SLC					
Null-centered magnitude correction	21.14	5.64	0.59	0.63	75.19

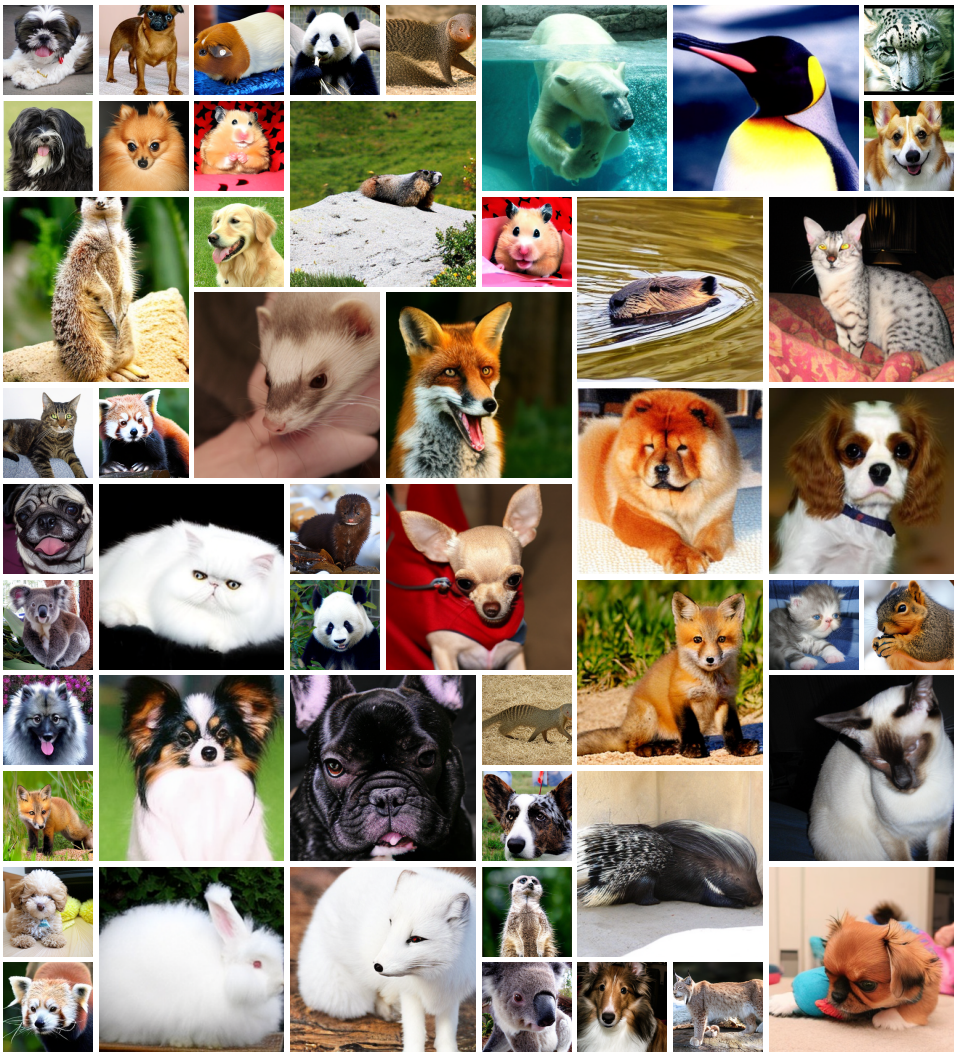


Figure 6: **The visualization results of SiT-XL/2 + CoVA use CFG with w = 4.0.**

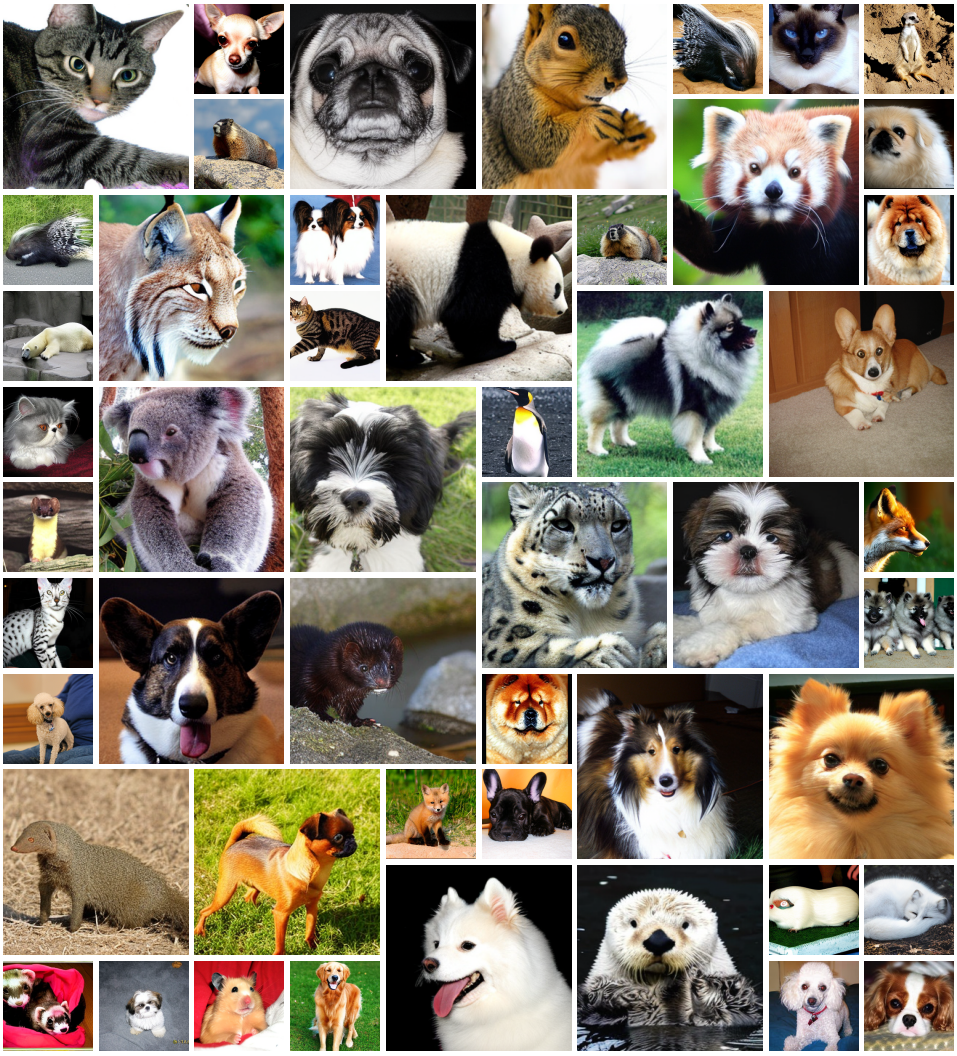


Figure 7: **The visualization results of SiT-XL/2 + CoVA use CFG with w = 4.0.**



Figure 8: **The visualization results of SiT-XL/2 + CoVA use CFG with w = 4.0.**

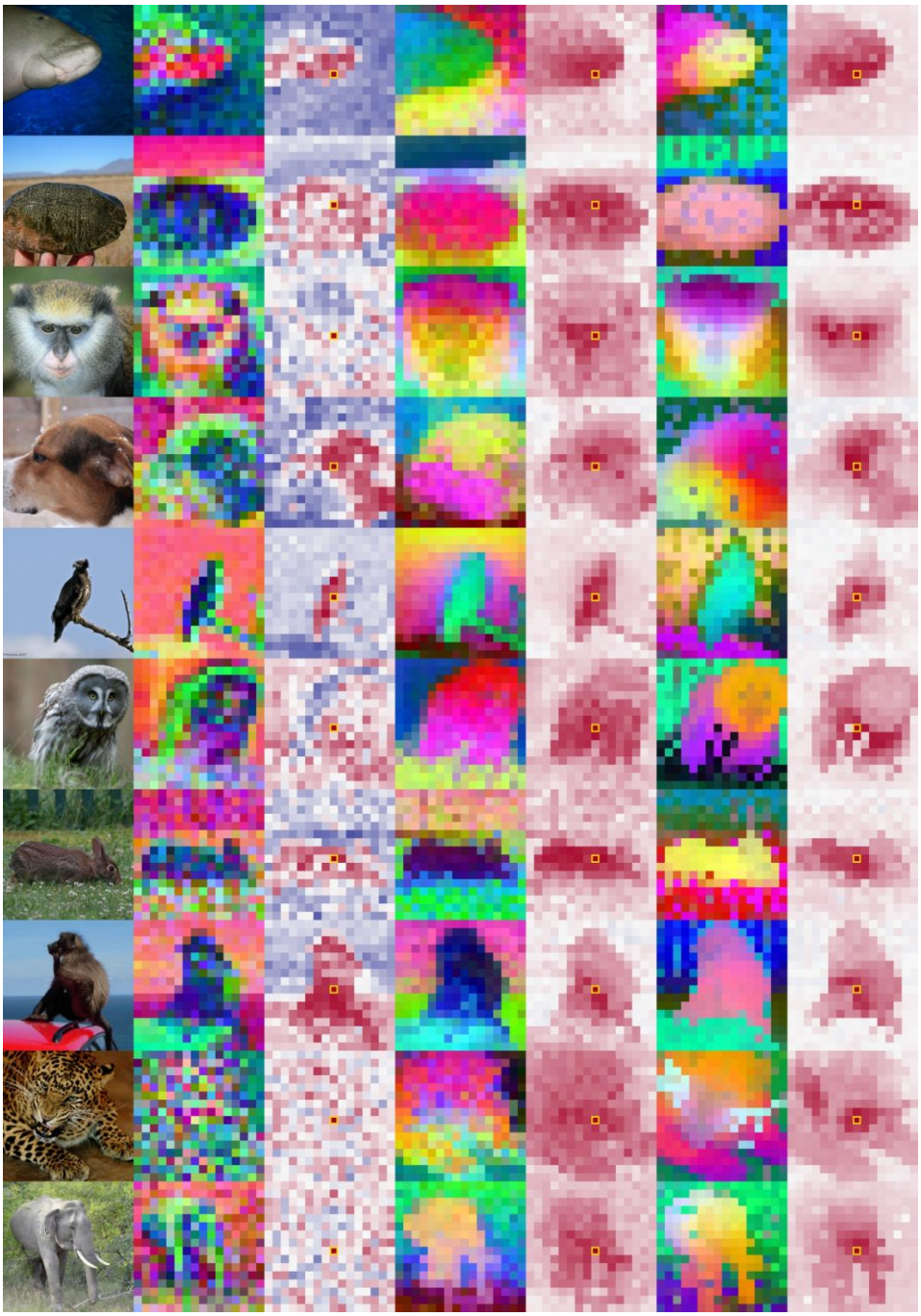


Figure 9: **Additional visualizations of spatial representations.**