V-Co: A Closer Look at Visual Representation Alignment via Co-Denoising

Han Lin[1], Xichen Pan[2], Zun Wang[1], Yue Zhang[1], Chu Wang[3], Jaemin Cho[4,5], and Mohit Bansal[1]

[1] University of North Carolina at Chapel Hill, USA
[2] New York University, USA
[3] Meta, USA
[4] AI2, USA
[5] Johns Hopkins University, USA
https://github.com/HL-hanlin/V-Co

Abstract. Pixel-space diffusion has recently re-emerged as a strong alternative to latent diffusion, enabling high-quality generation without pretrained autoencoders. However, standard pixel-space diffusion models receive relatively weak semantic supervision and are not explicitly designed to capture high-level visual structure. Recent representation-alignment methods (*e.g.*, REPA) suggest that pretrained visual features can substantially improve diffusion training, and *visual co-denoising* has emerged as a promising direction for incorporating such features into the generative process. However, existing co-denoising approaches often entangle multiple design choices, making it unclear which are truly essential. We therefore present **V-Co**, a systematic study of visual co-denoising in a unified JiT-based framework. This controlled setting allows us to isolate the ingredients that make visual co-denoising effective. Our study reveals two main ingredients. First, co-denoising benefits from preserving feature-specific computation while enabling flexible cross-stream interaction, which leads to a fully dual-stream architecture together with a structurally defined unconditional prediction for classifier-free guidance. Second, it requires both stronger semantic supervision and proper cross-stream calibration, which we realize through a perceptual-drifting hybrid loss and RMS-based feature rescaling. Together, these findings yield a simple recipe for visual co-denoising. Experiments on ImageNet-256 show that, at comparable model sizes, V-Co outperforms the underlying pixel-space diffusion baseline and strong prior pixel-diffusion methods while using fewer training epochs, offering practical guidance for future representation-aligned generative models.

Keywords: Diffusion Models · Representation Alignment · Co-Denoising

1 Introduction

Diffusion models [13, 27, 31] have achieved remarkable success in image generation. While much recent progress has been driven by latent diffusion models [32]

(LDMs), which denoise in compressed autoencoder spaces [21,32], an increasingly compelling alternative is pixel-space diffusion with scalable Transformer-based denoisers [6,24,26,29,45]. Recent systems such as JiT [24] show that direct pixel-space denoising can be competitive while avoiding autoencoder-induced biases and bottlenecks. However, pixel-level denoising objectives are not explicitly designed to enforce high-level semantic structure, making semantic representation learning and composition less sample-efficient.

In parallel, a growing body of work has explored how to inject external visual knowledge from strong pretrained encoders into diffusion training. One line of research adds *representation-alignment losses* that encourage diffusion features to match pretrained visual representations [33, 34, 37, 40, 46, 48]. Another performs denoising directly in a *representation latent space*, rather than in pixel or VAE latent space [3, 17, 35, 47]. A third line of work explores *joint generation or co-denoising* architectures, in which image latents are generated together with semantic features or other modalities so that the streams can exchange information throughout the denoising trajectory [1,2,4,8,9,14,18,22,41,43,49]. Among these directions, visual co-denoising provides a deeper form of integration by incorporating pretrained semantic representations directly into the denoising dynamics, rather than using them only as supervision or as an alternative latent space. However, existing co-denoising systems typically entangle multiple design choices (including architecture, guidance strategy, auxiliary supervision, and feature calibration), making it unclear which ingredients are most important and how to combine them into a robust and scalable recipe.

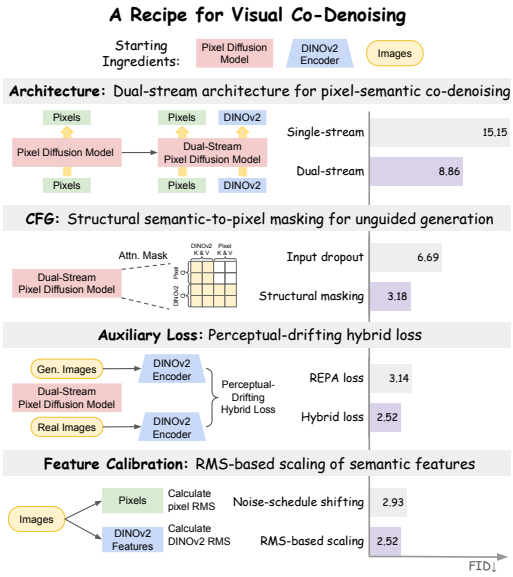

Fig. 1: Overview of V-Co.

In this paper, we study visual co-denoising as a mechanism for visual representation alignment. Rather than treating co-denoising as a fixed end-to-end design, we ask what makes it effective. To this end, we build a unified pixel-space testbed on top of JiT [24], where an image stream is jointly denoised with patch-level semantic features from a frozen pretrained visual encoder (*e.g.*, DINOv2 [30]). Within this controlled framework, we investigate four key questions: **(i)** what architecture best balances feature-specific processing and cross-stream interaction; **(ii)** how to define the unconditional branch for classifier-free guidance; **(iii)** which auxiliary objectives provide the most effective complementary

supervision; and **(iv)** how to calibrate semantic features relative to pixels during diffusion training. Our goal is not only to improve performance, but also to distill general principles for effective co-denoising.

Based on this study, we derive a simple yet effective **V**isual **Co**-Denoising (**V-Co**) recipe, illustrated in Fig. 1. First, from the perspective of model architecture, we show that effective visual co-denoising requires preserving feature-specific computation while enabling flexible cross-stream interaction. Among a broad range of shared-backbone and fusion-based variants, a *fully dual-stream* JiT consistently delivers the strongest performance (Sec. 3.2). Second, for classifier-free guidance (CFG), we introduce a novel *structural masking* formulation, where unconditional prediction is defined by explicitly masking the semantic-to-pixel pathway rather than by input-level corruption alone. This simple design proves substantially more effective than standard dropout-based alternatives in co-denoising (Sec. 3.3). Third, we observe that instance-level semantic alignment and distribution-level regularization play complementary roles, and leverage this insight to propose a novel *perceptual-drifting hybrid loss* that combines both within a unified objective, yielding the best generation quality in our study (Sec. 3.4). Finally, we show that RMS-based feature rescaling admits an equivalent interpretation as a semantic-stream noise-schedule shift via signal-to-noise ratio (SNR) matching, providing a simple and principled calibration rule for cross-stream co-denoising (Sec. 3.5). Together, these contributions transform visual co-denoising into a concrete recipe for visual representation alignment.

Empirically, V-Co yields strong gains on ImageNet-256 under the standard JiT [24] training protocol. Starting from a pixel-space JiT-B/16 backbone, our progressively improved recipe substantially outperforms both the original JiT baseline and prior co-denoising baselines (see Table 5), and achieves strong guided generation quality. Notably, V-Co-B/16 with only 260M parameters, matches JiT-L/16 with 459M parameters (FID 2.35 *vs*. 2.36). V-Co-L/16 and V-Co-H/16, trained for 500 and 300 epochs respectively, outperform JiT-G/16 with 2B parameters (FID 1.71 *vs*. 1.82) and other strong pixel-diffusion methods.

In summary, our contributions are three-fold:

- We present a principled study of visual representation alignment via co-denoising (**V-Co**) in pixel-space diffusion, systematically isolating the effects of architecture, CFG design, auxiliary objectives, and feature calibration.
- We introduce an effective recipe for visual co-denoising with two key innovations: *structural masking* for unconditional CFG prediction and a *perceptual-drifting hybrid loss* that combines instance-level alignment with distribution-level regularization. Our study further identifies a fully dual-stream architecture and RMS-based feature calibration as the preferred design choices.
- We show that these designs yield strong improvements on ImageNet-256 [10], outperforming the underlying pixel-space diffusion baseline (*i.e.*, JiT [24]) as well as prior pixel-space diffusion methods.

2 Related Work

Pixel-space diffusion generation. Recent work has shown that, with suitable architectural and optimization choices, diffusion models trained directly in pixel space can approach latent diffusion performance [32]. JiT [24] demonstrates that competitive pixel-space generation is possible with a minimalist Transformer design, while Simple Diffusion [15], PixelDiT [45], and HDiT [7] improve training and scalability. Other methods add stronger inductive biases, such as decomposition in DeCo [28] and perceptual supervision in PixelGen [29]. We adopt pixel-space diffusion rather than VAE-latent diffusion because it avoids autoencoder bottlenecks and learned latent-space biases, providing a cleaner setting for studying co-denoising and representation alignment.

Representation alignment for diffusion training. A growing line of work studies how pretrained visual representations can improve diffusion training. Recent analyses [33, 40] show that diffusion models learn meaningful internal features, but these are often weaker or less structured than those of strong self-supervised vision encoders. REPA [40] aligns intermediate diffusion features with pretrained representations such as DINOv2 [30], improving convergence and sample quality. Follow-up work studies which teacher properties matter most: iREPA [33] highlights spatial structure, while REPA-E [23] extends REPA-style supervision to end-to-end latent diffusion training with the VAE. Recent results also suggest that REPA-style alignment is most beneficial early in training and may over-constrain the representation space if applied too rigidly [40]. Motivated by this, we study representation alignment through co-denoising, compare auxiliary objectives beyond REPA, and introduce a stronger hybrid alternative.

Visual co-denoising and joint generation across modalities. Recent work has increasingly explored *joint denoising* or *joint generation* of multiple signals to improve information transfer, controllability, and structural consistency. In image generation, Latent Forcing [1] and ReDi [22] jointly model image latents and semantic features. In video generation, VideoJAM [4], UDPDiff [42], and UnityVideo [18] jointly generate video with structured signals such as segmentation, depth, or flow. Similar ideas extend to audio–visual generation [41], robotics and world modeling [2, 8, 9, 43, 49], and multimodal sequence modeling [14]. In contrast to these task-specific end-to-end designs, we provide a controlled study of visual co-denoising itself, isolating the architectural, guidance, loss, and calibration choices that make it effective and distilling them into a practical recipe for visual representation alignment.

3 A Closer Look at Visual Co-Denoising

In this section, we first formalize visual co-denoising in Section 3.1, then conduct a systematic study of the key design choices that govern its effectiveness, including model architecture (Section 3.2), unconditional prediction for CFG (Section 3.3), auxiliary training objectives (Section 3.4), and feature calibration via rescaling (Section 3.5). Starting from a standard pixel-space diffusion baseline

(*e.g.*, JiT [24]), we use controlled ablations to isolate each component's contribution and derive a practical recipe, introducing *new designs tailored for visual co-denoising* along the way. Experiment setup details and additional ablations are deferred to Appendix Sec. A and Appendix Sec. B respectively.

3.1 Co-Denoising Formulation

We formalize visual co-denoising within a unified framework. Unlike standard pixel-space diffusion, which denoises only the image stream, co-denoising introduces an additional semantic feature stream from a pretrained visual encoder (*e.g.*, DINOv2 [30]). The core idea is to jointly denoise the pixel and semantic streams under a shared diffusion process, allowing the semantic stream to provide complementary supervision for semantically richer generation.

All experiments in this section follow the JiT [24] ablation protocol on ImageNet 256×256 [10], training JiT-B/16 for 200 epochs. Unless otherwise specified, we adopt the JiT training configuration without hyperparameter tuning.

Concretely, we extend the x-prediction and v-loss formulation of JiT to jointly denoise pixels and pretrained semantic features. Let $\boldsymbol{x}$ denote the clean image and $\boldsymbol{d}$ denote its encoded patch-level semantic features. We sample independent Gaussian noise $\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_d \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for the two streams. At diffusion time $t \in [0, 1]$, the corresponding noised inputs are

$$\boldsymbol{z}_t^x = t\,\boldsymbol{x} + (1-t)\,\boldsymbol{\epsilon}_x, \qquad \boldsymbol{z}_t^d = t\,\boldsymbol{d} + (1-t)\,\boldsymbol{\epsilon}_d. \tag{1}$$

Given $(\boldsymbol{z}_t^x, \boldsymbol{z}_t^d, t, c)$, where c denotes the class condition, the co-denoising model jointly predicts the clean targets for the pixel and semantic streams:

$$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{d}}) = f_{\boldsymbol{\theta}}(\boldsymbol{z}_t^x, \boldsymbol{z}_t^d, t, c), \tag{2}$$

where $f_{\boldsymbol{\theta}}$ denotes the co-denoising model, which could be implemented as either a shared-backbone or dual-stream architecture depending on the design variant. Following JiT, we convert these clean predictions into velocity predictions,

$$\hat{\boldsymbol{v}}_x = (\hat{\boldsymbol{x}} - \boldsymbol{z}_t^x)/(1-t), \qquad \hat{\boldsymbol{v}}_d = (\hat{\boldsymbol{d}} - \boldsymbol{z}_t^d)/(1-t), \tag{3}$$

and supervise them using the corresponding ground-truth velocities,

$$\boldsymbol{v}_x = \boldsymbol{x} - \boldsymbol{\epsilon}_x = (\boldsymbol{x} - \boldsymbol{z}_t^x)/(1-t), \qquad \boldsymbol{v}_d = \boldsymbol{d} - \boldsymbol{\epsilon}_d = (\boldsymbol{d} - \boldsymbol{z}_t^d)/(1-t). \tag{4}$$

The final objective is a weighted sum of the pixels and semantic features v-losses:

$$\mathcal{L}_{\text{v-co}} = \mathbb{E}\Big[\|\hat{\boldsymbol{v}}_x - \boldsymbol{v}_x\|_2^2 + \lambda_d \,\|\hat{\boldsymbol{v}}_d - \boldsymbol{v}_d\|_2^2\Big], \tag{5}$$

where λ_d controls the weight of the semantic stream. This formulation provides a unified testbed for studying the effects of architecture, guidance, auxiliary objectives, and feature calibration on representation alignment in co-denoising.

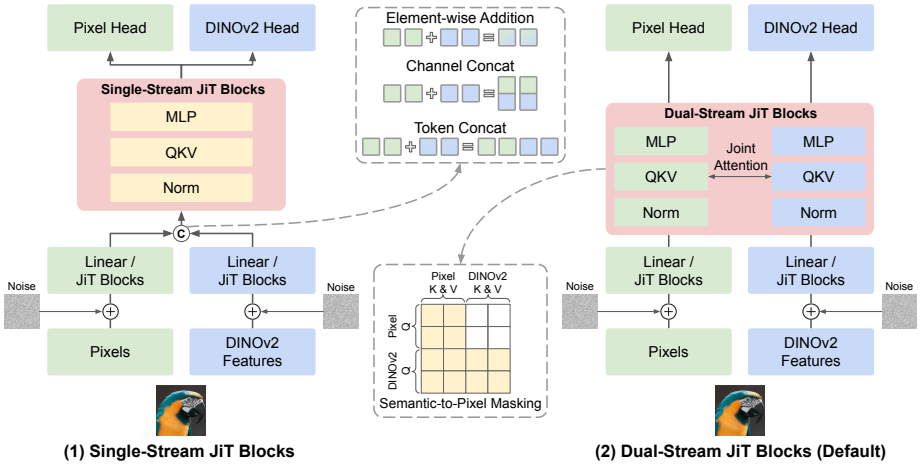

Fig. 2: Single-stream and dual-stream architectures for visual co-denoising. In the *single-stream design* (left), noised pixels and DINOv2 features are fused after lightweight stream-specific preprocessing and then processed by shared JiT blocks. We study direct addition, channel concatenation, and token concatenation (see Sec. 3.2). In the *dual-stream design* (right), the two streams use separate normalization, MLP, and attention projections, while interacting through joint self-attention. A semantic-to-pixel attention mask is used to define the unconditional prediction for CFG (see Sec. 3.3). Both designs use separate output heads for pixel and DINOv2 prediction.

3.2 What Architecture Best Supports Visual Co-Denoising?

We begin by studying how semantic features should be integrated into a pixel-space diffusion backbone for co-denoising. Our goal is to identify the *architectural design that most effectively transfers information from pretrained semantic visual encoders to pixel features without limiting the expressiveness of the diffusion model*. To this end, we compare lightweight fusion within a largely shared backbone against more expressive designs that preserve feature-specific processing while enabling controlled cross-stream interaction. Fig. 2 illustrates the architectural variants, and Table 1 summarizes the corresponding results.

Baselines. We first report results for the original JiT-B/16 backbone [24] and a widened variant that increases the hidden dimension from 768 to 1088, so as to match the parameter count of the dual-stream models introduced later. We also include Latent Forcing [1] as a representative co-denoising baseline. For fair comparison, we keep the number of JiT blocks traversed by the pixel stream fixed across all variants, and maintain this setting throughout this subsection.

Single-stream variants. We consider a shared-backbone setting (Fig. 2, left) where pixel tokens $\boldsymbol{x}$ and semantic tokens $\boldsymbol{d}$ share most parameters. Within this setting, we compare three fusion strategies with Latent Forcing [1] architecture:

– **Direct Addition** (row (d)): Pixel tokens $\boldsymbol{x} \in \mathbb{R}^{n \times d_1}$ and semantic features $\boldsymbol{d} \in \mathbb{R}^{n \times d_2}$ are first projected into a shared hidden space $\mathbb{R}^{n \times d}$ via lightweight

Table 1: Comparison of architectural designs for visual co-denoising. We compare baseline backbones, single-stream fusion strategies, and dual-stream fusion variants with different allocations of feature-specific and shared/dual-stream blocks. All variants keep the pixel stream depth fixed at 12 JiT blocks for fair comparison. JiT-B/16‡ and JiT-B/16† denote widened variants with hidden dimensions increased from 768 to 1024 and 1088, respectively, to match the parameter counts of the dual-stream models. Blue rows mark the stronger variants used in subsequent analysis. Following previous works [1, 24], we mainly use FID as reference.

	Model	Backbone	#Params	#Feature-Specific Blocks	#Shared/Dual-Stream Blocks	CFG=1.0 FID↓	CFG=1.0 IS↑
Baselines							
(a)	JiT-B/16 [24]	JiT-B/16	133M	-	-	32.54	49.5
(b)	JiT-B/16 [24]	JiT-B/16†	261M	-	-	22.67	69.9
(c)	LatentForcing [1]	JiT-B/16	156M	2	10	13.06	102.2
Single-Stream JiT Architecture							
(d)	DirectAddition	JiT-B/16	156M	2	10	15.15	103.4
(e)	ChannelConcat	JiT-B/16	157M	2	10	14.33	107.7
(f)	TokenConcat	JiT-B/16	156M	2	10	14.70	103.8
(g)	TokenConcat	JiT-B/16	177M	4	8	12.59	112.8
(h)	TokenConcat	JiT-B/16	198M	6	6	12.35	116.7
(i)	TokenConcat	JiT-B/16‡	265M	6	6	9.74	129.45
Dual-Stream JiT Architecture							
(j)	TokenConcat	JiT-B/16	260M	6	6	11.78	115.4
(k)	TokenConcat	JiT-B/16	260M	4	8	11.40	118.3
(l)	TokenConcat	JiT-B/16	260M	2	10	10.24	124.5
(m)	TokenConcat	JiT-B/16	260M	0	12	**8.86**	**132.8**

linear layers, then fused by *element-wise addition* and passed through shared JiT blocks. The pixel and semantic streams have two separate output heads.

- **Channel-concatenation fusion** (row (e)): Pixel tokens $\boldsymbol{x}$ and semantic features $\boldsymbol{d}$ are concatenated along the channel dimension $\mathbb{R}^{n\times(d_1+d_2)}$, and then linearly projected to the hidden dimension $\mathbb{R}^{n\times d}$ of JiT blocks.
- **Token-concatenation fusion** (rows (f-i)): Instead of concatenating along the channel dimension, we concatenate $\boldsymbol{x}$ and $\boldsymbol{d}$ tokens along the sequence dimension $\mathbb{R}^{2n\times d}$ and input the combined token sequence into the JiT blocks.

Dual-stream variants. Motivated by the limitations of heavily shared backbones, we further introduce a *dual-stream* JiT architecture, illustrated on the right of Fig. 2, in which the pixel and semantic streams maintain separate normalization layers, MLPs, and attention projections (*i.e.*, Q/K/V), while interacting through joint self-attention. This design allows the model to adaptively determine *where* and *how* the two streams interact, while preserving dedicated processing pathways for each stream.

Analysis. As shown in Table 1, token-concatenation fusion outperforms direct addition and channel concatenation among the single-stream variants (rows (d)–(f)), suggesting that preserving feature-specific representations before interaction is preferable to early fusion in a shared space. Moreover, within token-concatenation, allocating more blocks to feature-specific processing consistently improves performance (rows (f)–(h)), indicating that excessive parameter sharing limits the model's ability to preserve semantic information. Finally, among

Table 2: Comparison of unconditional prediction designs for classifier-free guidance under co-denoising. We report unguided results at CFG= 1.0 and guided results at CFG= 2.9, which is the default guided evaluation setting in JiT [24].

Uncond. Type	Cond. Pred.	Uncond. Pred.	CFG=1.0		CFG=2.9	
			FID↓	IS↑	FID↓	IS↑
Independently Drop Labels & Semantic Features						
(a) Zero Embedding	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \mathbf{0}], \emptyset, t)$	9.17	126.3	6.69	165.6
(b) Learnable `[null]` Token	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \texttt{[null]}], \emptyset, t)$	9.37	126.7	6.64	165.7
(c) Bidirectional Cross-Stream Mask	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], \emptyset, t)$	11.08	101.9	7.17	143.5
(d) Semantic-to-Pixel Mask	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], \emptyset, t)$	**7.28**	**136.8**	**3.59**	**189.6**
Jointly Drop Labels & Semantic Features						
(e) Zero Embedding	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \mathbf{0}], \emptyset, t)$	15.58	98.8	24.75	82.4
(f) Learnable `[null]` Token	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \texttt{[null]}], \emptyset, t)$	10.80	118.3	25.2	88.3
(g) Bidirectional Cross-Stream Mask	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], \emptyset, t)$	7.53	129.1	5.66	173.6
(h) Semantic-to-Pixel Mask	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], y, t)$	$f_{\boldsymbol{\theta}}([\boldsymbol{x}_t, \boldsymbol{d}_t], \emptyset, t)$	**5.62**	**158.5**	**3.18**	**219.4**

the dual-stream variants, the fully dual-stream architecture (row (m)) achieves the best FID of 8.86 under a comparable number of trainable parameters (row (i) and rows (j)–(l)), showing that allowing the model to *dynamically learn* cross-stream interaction at each block is more effective than imposing a fixed interaction pattern through a largely shared backbone. Therefore, we adopt the fully dual-stream architecture as the default model design in the remaining analysis.

Finding 1: Effective visual feature alignment requires preserving feature-specific processing while enabling flexible cross-stream interaction; token concatenation and a fully dual-stream JiT emerge as the preferred designs.

3.3 How to Define Unconditional Prediction for CFG?

To enable classifier-free guidance (CFG), the model must define an unconditional prediction, *i.e.*, a prediction in which the conditioning signals are removed. In our co-denoising setting, this is nontrivial because the model is conditioned on both class labels and semantic features. Guided sampling combines the conditional and unconditional predictions in the pixel and semantic streams as

$$\hat{\boldsymbol{v}}_x = \hat{\boldsymbol{v}}_x^{\text{uncond}} + s\left(\hat{\boldsymbol{v}}_x^{\text{cond}} - \hat{\boldsymbol{v}}_x^{\text{uncond}}\right), \quad \hat{\boldsymbol{v}}_d \quad = \hat{\boldsymbol{v}}_d^{\text{uncond}} + s\left(\hat{\boldsymbol{v}}_d^{\text{cond}} - \hat{\boldsymbol{v}}_d^{\text{uncond}}\right) \quad (6)$$

where s denotes the CFG scale. Since guided generation depends critically on the quality of the unconditional branch, we next investigate *how to define an effective unconditional prediction for CFG in the co-denoising setting.*

Input-dropout baselines. Following prior work [1, 22], we first consider baseline unconditional predictions that drop conditioning inputs (semantic features and class labels) during training. Specifically, for semantic feature dropping, we use either **(1)** zeros or **(2)** a learnable `[null]` token to replace the semantic features. For each choice, we compare independent dropout of the class label and semantic features (rows (a)–(b)) against joint dropout (rows (e)–(f)).

Attention mask between pixel and semantic features. Beyond input-level dropout, we leverage the dual-stream architecture to define a *structurally unconditional* pathway. For unconditional samples, we apply *semantic-to-pixel masking* (see Fig. 3), which blocks cross-stream attention from the semantic stream to the pixel stream so that the pixel branch receives no semantic conditioning signal (rows (d) and (h)). We also study a symmetric variant, *bidirectional cross-stream masking*, which blocks attention in both directions (rows (c) and (g)). These variants test whether unconditional prediction is better defined through explicit control of information flow rather than input-level corruption.

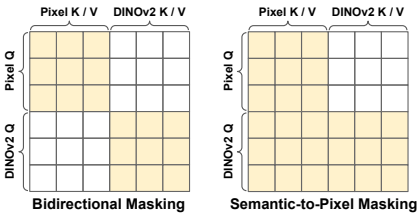

Fig. 3: Comparison of two attention-masking strategies.

Analysis. Table 2 first shows that under the baseline input-dropout strategy, independently dropping the class label and semantic features (rows (a)–(b)) performs substantially better than jointly dropping them (rows (e)–(f)). We hypothesize that jointly dropping both conditions makes the pixel-space guidance direction, $\Delta_x = \hat{\boldsymbol{v}}_x^{\text{cond}} - \hat{\boldsymbol{v}}_x^{\text{uncond}}$, a poorly calibrated estimate of the desired conditional guidance signal, which is then amplified by CFG scaling. In contrast, independent dropout exposes the model to partially conditioned cases and thus appears to improve the robustness of the learned guidance direction.

More importantly, explicitly defining the unconditional pathway through *structural* masking (rows (c)–(d)) is markedly more effective than input-level dropout (rows (a)–(b)) under independent dropout, suggesting that blocking semantic information from reaching the pixel branch yields a more reliable unconditional prediction. Among the structural variants, masking only the semantic-to-pixel pathway (row (d)) performs best, indicating that unconditional generation only requires removing semantic influence on the pixel output, while preserving the reverse pixel-to-semantic interaction remains beneficial. For structural masking, jointly dropping labels and semantic features (rows (g)–(h)) outperforms independent dropout (rows (c)–(d)), suggesting that once the unconditional branch is defined structurally, removing all conditioning sources during training better matches inference-time behavior.

Finding 2: For CFG under co-denoising, the most effective unconditional design is structural semantic-to-pixel masking with joint dropout of conditioning signals during training.

3.4 Which Auxiliary Loss Best Improves Co-Denoising?

The default V-Co objective in Eq. (5) supervises both streams through the co-denoising v-loss, but it mainly enforces local target matching and may not fully capture higher-level semantic alignment. We therefore study *which auxiliary objectives provide the most effective complementary supervision in representation space*, and ultimately design a hybrid loss that better combines their strengths.

Balancing pixel and semantic losses. Before exploring additional auxiliary objectives, we first tune the relative weight of the semantic-stream loss through λ_d in Eq. (5). As shown in Fig. 4, $\lambda_d \in \{0.01, 0.1\}$ gives the best FID. Under these settings, the average parameter-gradient norm in the pixel branch is approximately $4\times$ and $2\times$ that of the semantic branch, respectively. This suggests that semantic supervision is most effective when it provides meaningful guidance while remaining secondary to the primary pixel-space objective.

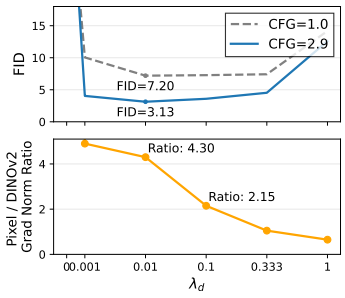

Fig. 4: Influence of λ_d.

REPA loss [23]. On top of the co-denoising objective, we further consider a REPA-style representation alignment loss on the pixel branch. Concretely, let $\boldsymbol{h}_\ell^x$ denote the intermediate hidden representation of the pixel branch at layer ℓ. We align this hidden state to the representation of the ground-truth image $\boldsymbol{x}$ extracted by a frozen pretrained DINOv2 visual encoder $\phi(\cdot)$. The auxiliary objective is defined as $\mathcal{L}_{\text{REPA}} = \left\| g(\boldsymbol{h}_\ell^x) - \phi(\boldsymbol{x}) \right\|_2^2$, where $g(\cdot)$ denotes a lightweight MLP projector used to map the intermediate hidden state to the encoder feature space.

Perceptual loss in semantic feature space. We also consider a perceptual loss [20, 29] in the pretrained semantic feature space. Specifically, given the predicted clean image $\hat{\boldsymbol{x}}$ and the ground-truth image $\boldsymbol{x}$, we extract their features using a frozen pretrained DINOv2 visual encoder $\phi(\cdot)$ and minimize their discrepancy: $\mathcal{L}_{\text{perc}} = \|\phi(\hat{\boldsymbol{x}}) - \phi(\boldsymbol{x})\|_2^2$. Unlike REPA, which aligns intermediate hidden states, this loss directly supervises the predicted image in semantic feature space.

Drifting loss in semantic feature space. Drifting loss [11] was recently proposed for single-step image generation to move the generated distribution toward the real one. Unlike diffusion v-prediction, REPA, and perceptual losses, which impose pairwise supervision between generated and real samples, drifting loss operates at the distribution level. To study its effectiveness in multi-step diffusion, we implement it in DINO feature space. Let $\phi(\cdot)$ denote a frozen DINOv2 encoder, let $\hat{\boldsymbol{x}}$ be the predicted clean image from the pixel branch, and define $\boldsymbol{u} = \phi(\hat{\boldsymbol{x}})$. We construct a drifting field as follows:

$$V(\boldsymbol{u}) = V^+(\boldsymbol{u}) - V^-(\boldsymbol{u}), \tag{7}$$

$$V^+(\boldsymbol{u}) = \frac{1}{Z_+(\boldsymbol{u})} \mathbb{E}_{\boldsymbol{x}^+ \sim p_{\text{data}}} \left[k(\boldsymbol{u}, \phi(\boldsymbol{x}^+))(\phi(\boldsymbol{x}^+) - \boldsymbol{u}) \right], \tag{8}$$

$$V^-(\boldsymbol{u}) = \frac{1}{Z_-(\boldsymbol{u})} \mathbb{E}_{\boldsymbol{x}^- \sim p_{\text{gen}}} \left[k(\boldsymbol{u}, \phi(\boldsymbol{x}^-))(\phi(\boldsymbol{x}^-) - \boldsymbol{u}) \right], \tag{9}$$

where p_{data} and p_{gen} denote the real and generated image distributions respectively. $k(\boldsymbol{a}, \boldsymbol{b}) = \exp(-\|\boldsymbol{a} - \boldsymbol{b}\|_2^2 / \tau)$ is a similarity kernel and Z_+, Z_- normalize the kernel weights. The drifting loss is defined as $\mathcal{L}_{\text{drift}} = \|\boldsymbol{u} - \text{sg}(\boldsymbol{u} + V(\boldsymbol{u}))\|_2^2$, where $\text{sg}(\cdot)$ denotes stop-gradient.

Table 3: Comparison of auxiliary losses applied to V-Co. We report unguided results at CFG= 1.0 and guided results at the best CFG selected from a sweep over [1.5, 5.5] for each method. All results here are obtained after 300 epochs of training.

Aux. Loss Type	Unguided (CFG=1.0)		Guided (Best CFG> 1.0)	
	FID↓	IS↑	FID↓	IS↑
Baseline: V-Co w/ default V-Pred Loss	5.38	153.6	2.96	206.6
(a) V-Co + REPA Loss	5.63	149.4	2.91 (0.05 ↓)	202.8
(b) V-Co + Perceptual Loss	**4.28**	177.6	2.73 (0.23 ↓)	228.5
(c) V-Co + Drifting Loss	4.86	164.3	2.85 (0.11 ↓)	211.5
(d) V-Co + Perceptual-Drifting Hybrid Loss	4.44	**189.0**	**2.44** (0.52 ↓)	**249.9**

Perceptual-Drifting Hybrid loss. Perceptual loss and drifting loss provide two complementary forms of supervision. Perceptual loss encourages *instance-level* semantic fidelity by pulling each generated image toward the semantic feature of its paired ground-truth target, while drifting loss promotes *distributional coverage* by discouraging generated features from collapsing toward dense regions of the generated distribution. Motivated by this complementarity, we propose a hybrid objective that formulates perceptual alignment as a positive vector field and drifting-based repulsion as a negative correction.

Unlike the original drifting loss, we replace its positive real-distribution term with the current sample's *paired perceptual field* while retaining the generated-distribution term as a negative correction. This is better suited to multi-step denoising, where each noisy input is paired with a specific ground-truth image.

Specifically, we define the positive perceptual field as:

$$V_{\text{pos}}(\boldsymbol{u}_i) = \phi(\boldsymbol{x}_i) - \phi(\hat{\boldsymbol{x}}_i), \tag{10}$$

which pulls the generated sample toward the semantic feature of its paired ground-truth target. The negative field computes repulsion from nearby generated samples within the same class. For each sample i, we compute normalized kernel weights over other samples $j \neq i$ of the same class:

$$\alpha_{ij} = \frac{\exp\left(-\|\phi(\hat{\boldsymbol{x}}_i) - \phi(\hat{\boldsymbol{x}}_j)\|^2/\tau_{\text{rep}}\right)}{\sum_{k\neq i}\exp\left(-\|\phi(\hat{\boldsymbol{x}}_i) - \phi(\hat{\boldsymbol{x}}_k)\|^2/\tau_{\text{rep}}\right)}, \tag{11}$$

where τ_{rep} is the repulsion temperature. Note that $\sum_{j\neq i}\alpha_{ij} = 1$. The repulsion direction points from sample i toward the weighted centroid of its neighbors:

$$V_{\text{neg}}(\boldsymbol{u}_i) = \sum_{j\neq i}\alpha_{ij}\phi(\hat{\boldsymbol{x}}_j) - \phi(\hat{\boldsymbol{x}}_i). \tag{12}$$

To adaptively balance attraction and repulsion, we introduce a *similarity-based gating* mechanism based on how close the generated feature is to its target:

$$s_i = \exp\left(-\frac{\|\phi(\hat{\boldsymbol{x}}_i) - \phi(\boldsymbol{x}_i)\|^2}{\tau_{\text{gate}}}\right), \tag{13}$$

Table 4: Feature rescaling *vs*. noise-schedule shifting in V-Co. We compare our default V-Co model with RMS scaling against variants *without RMS scaling* and *with noise-schedule shifting*. We report unguided results at CFG= 1.0 and guided results at the best CFG selected from a sweep over [1.5, 5.5] for each method.

Model	Unguided (CFG=1.0)		Guided (Best CFG> 1.0)	
	FID↓	IS↑	FID↓	IS↑
V-Co w/o rms scaling	9.12	188.6	5.28	150.4
V-Co w/o rms scaling + noise schedule shifting	**4.81**	**205.6**	2.93	**272.4**
V-Co w/ rms scaling (default)	5.38	177.0	**2.52**	242.6

where τ_{gate} is a temperature parameter controlling the sensitivity of the gate. We then combine the two fields into a hybrid field:

$$V_{\text{hyb}}(\boldsymbol{u}_i) = s_i \cdot V_{\text{pos}}(\boldsymbol{u}_i) - (1 - s_i) \cdot V_{\text{neg}}(\boldsymbol{u}_i). \tag{14}$$

Intuitively, when the generated feature is far from the target ($s_i \approx 0$), repulsion dominates to prevent mode collapse; when the generated feature is close to the target ($s_i \approx 1$), pure attraction ensures clean convergence.

The final objective is:

$$\mathcal{L} = \mathcal{L}_{\text{v-co}} + \lambda_{\text{hyb}}\, \mathcal{L}_{\text{hyb}} = \mathcal{L}_{\text{v-co}} + \lambda_{\text{hyb}} \left\| \boldsymbol{u}_i - \text{sg}\big(\boldsymbol{u}_i + V_{\text{hyb}}(\boldsymbol{u}_i)\big) \right\|_2^2. \tag{15}$$

Analysis. Table 3 and Fig. 5 quantify the impact of different auxiliary objectives on co-denoising. REPA provides only a marginal guided improvement over the default V-Co objective (FID 2.91 vs. 2.96 for CFG> 1), suggesting limited benefit from intermediate alignment and potential constraints on model's expressiveness [40]. Perceptual loss yields a larger gain (FID 2.73), indicating effective instance-level semantic alignment, while drifting loss provides a smaller improvement (FID 2.85) but contributes distribution-level supervision. Combining these complementary signals, our hybrid objective achieves the best guided result (FID 2.44), suggesting that instance-level alignment is most effective when paired with explicit distribution-level correction.

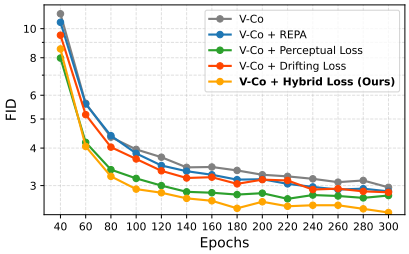

Fig. 5: Comparison of guided FID.

Finding 3: For visual co-denoising, auxiliary losses work best when combining instance-level alignment with distribution-level regularization.

3.5 How Should Semantic Features Be Calibrated for Co-Denoising?

Before concluding the recipe, we consider *how to calibrate the semantic stream relative to the pixel stream during co-denoising*. Since the two streams lie in dif-

ferent representation spaces and can have very different signal scales, using the same diffusion timestep may induce mismatched denoising difficulty, leading to imbalanced optimization. This motivates two natural strategies: rescaling the semantic features to match the pixel-space signal level, or equivalently shifting the semantic-stream diffusion schedule [1]. As we show below, these two formulations are equivalent in signal-to-noise ratio (SNR).

SNR matching via feature rescaling. As defined in Eq. (1), pixels and semantic features share the same timestep $t \in [0, 1]$. Under this flow-matching parameterization, denoising difficulty is governed by:

$$\mathrm{SNR}(t) \propto \frac{t^2\,\mathbb{E}\|\boldsymbol{s}\|^2}{(1-t)^2\,\mathbb{E}\|\boldsymbol{\epsilon}\|^2}, \tag{16}$$

where $\boldsymbol{s}$ is the clean signal and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is the injected noise. Since t is shared and the noise scale is fixed, matching denoising difficulty reduces to matching signal scale. We therefore rescale semantic features to match the pixel SNR:

$$\boldsymbol{d}' = \alpha \boldsymbol{d}, \qquad \alpha = \frac{\mathrm{RMS}_{\boldsymbol{x}}}{\mathrm{RMS}_{\boldsymbol{d}}}. \tag{17}$$

Equivalent SNR matching via timestep shifting. Equivalently, one can keep the semantic feature $\boldsymbol{d}$ fixed and instead shift its timestep from t to t' such that $\mathrm{SNR}_{\boldsymbol{d}}(t') = \mathrm{SNR}_{\boldsymbol{d}'}(t)$. Using Eq. (16), this gives $\frac{t'}{1-t'} = \alpha\,\frac{t}{1-t}$, and thus

$$t' = \frac{\alpha t}{1 + (\alpha - 1)t}. \tag{18}$$

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

Analysis. In Table 4, we compare our default V-Co model against variants *without RMS scaling* and *with noise-schedule shifting* in place of RMS scaling. Removing RMS scaling substantially worsens performance relative to our default setting (guided FID: 5.28 *vs*. 2.52). Replacing RMS scaling with noise-schedule shifting gives broadly comparable overall results, but yields worse guided FID (2.93 *vs*. 2.52), consistent with the SNR-based equivalence discussed above. In practice, we adopt RMS scaling because of its simplicity and strong performance.

Finding 4: Effective co-denoising requires calibrating the semantic and pixel streams to comparable denoising difficulty. RMS-based feature rescaling provides a simple and practical solution.

4 Full Recipe and SoTA Comparison

We combine the best-performing ablation choices into a practical visual co-denoising (**V-Co**) recipe: a fully dual-stream JiT backbone (Sec. 3.2), structural semantic-to-pixel masking with joint dropout for CFG (Sec. 3.3), a perceptual-drifting hybrid loss (Sec. 3.4), and RMS-based feature rescaling (Sec. 3.5). These

Table 5: Reference results on ImageNet 256×256. FID and IS of 50K samples are evaluated. The "pre-training" columns list the external models required to obtain the results. The #Params column reports the parameter count of the denoising model and the decoder (if used). Following previous works [1, 24], we mainly use FID as reference.

ImageNet 256 × 256	Pre-training		#Params	#Epochs	FID↓	IS↑
	Tokenizer	SSL Encoder				
Latent-space Diffusion						
DiT-XL/2 [31]	SD-VAE	-	675+49M	1400	2.27	278.2
SiT-XL/2 [27]	SD-VAE	-	675+49M	1400	2.06	277.5
REPA [40], SiT-XL/2	SD-VAE	DINOv2	675+49M	800	1.42	305.7
LightningDiT-XL/2 [44]	VA-VAE	DINOv2	675+49M	800	1.35	295.3
DDT-XL/2 [39]	SD-VAE	DINOv2	675+49M	400	1.26	310.6
RAE [47], DiT^{DH}-XL/2	RAE	DINOv2	839+415M	800	**1.13**	262.6
Pixel-space (non-diffusion)						
JetFormer [36]	-	-	2.8B	-	6.64	-
FractalMAR-H [25]	-	-	848M	600	6.15	**348.9**
Pixel-space Diffusion						
ADM-G [12]	-	-	554M	400	4.59	186.7
RIN [19]	-	-	410M	-	3.42	182.0
SiD [15], UViT/2	-	-	2B	800	2.44	256.3
VDM++, UViT/2	-	-	2B	800	2.12	267.7
SiD2 [16], UViT/2	-	-	N/A	-	1.73	-
SiD2 [16], UViT/1	-	-	N/A	-	1.38	-
PixelFlow [5], XL/4	-	-	677M	320	1.98	282.1
PixNerd [38], XL/16	-	DINOv2	700M	160	2.15	297.0
DeCo-XL/16 [28]	-	DINOv2	682M	600	1.69	304.0
PixelGen-XL/16 [29]	-	DINOv2	676M	160	1.83	293.6
ReDi [22], SiT-XL/2	-	DINOv2	675M	350	1.72	278.7
ReDi [22], SiT-XL/2	-	DINOv2	675M	800	1.61	295.1
Latent Forcing [1], JiT-B/16	-	DINOv2	465M	200	2.48	-
JiT-B/16 [24]	-	-	131M	600	3.66	275.1
JiT-L/16 [24]	-	-	459M	600	2.36	298.5
JiT-H/16 [24]	-	-	953M	600	1.86	303.4
JiT-G/16 [24]	-	-	2B	600	1.82	292.6
V-Co-B/16	-	DINOv2	260M	200	2.52	242.6
V-Co-L/16	-	DINOv2	918M	200	2.10	243.0
V-Co-H/16	-	DINOv2	1.9B	200	1.85	246.5
V-Co-B/16	-	DINOv2	260M	600	2.35	261.1
V-Co-L/16	-	DINOv2	918M	500	1.72	245.3
V-Co-H/16	-	DINOv2	1.9B	300	1.71	263.3

components are complementary: the architecture determines *where* streams interact, masking defines *how* guidance is formed, the hybrid loss specifies *what* semantic supervision to apply, and RMS scaling matches denoising difficulty.

We next evaluate this full recipe against prior SoTA methods on ImageNet [10] 256 × 256, including latent-space diffusion models, and pixel-space methods. As shown in Table 5, V-Co achieves strong performance among pixel-space diffusion models. Notably, V-Co-B/16 with only 260M parameters, matches JiT-L/16 with 459M parameters (FID 2.35 *vs*. 2.36). V-Co-L/16 and V-Co-H/16, trained for 500 and 300 epochs respectively, outperform JiT-G/16 with 2B parameters (FID 1.71 *vs*. 1.82) and other strong pixel-diffusion methods. In addition, our method with simple RMS scaling matches or outperforms Latent Forcing [1], which relies on separate noise schedules for pixels and DINOv2 features. These

Table 6: Computational efficiency comparison with JiT. We report trainable parameters, per-sample training GFLOPs, an epoch-normalized training-compute proxy, and inference throughput. For V-Co, total training GFLOPs include both the denoising Transformer and the frozen DINOv2 encoder used to extract semantic features during training. Inference throughput is measured without the DINOv2 encoder, since V-Co does not require the external encoder at sampling time.

Model	#Params	Train GFLOPs			Imgs/sec↑	Epochs / FID↓
		DiT	DINO Enc.	Total↓		
JiT-L/16 [24]	459M	89	-	89	0.329	600 / 2.36
V-Co-B/16	260M	53	35	88	**0.484**	600 / **2.35**
JiT-H/16 [24]	953M	183	-	183	0.175	600 / 1.86
JiT-G/16 [24]	2.0B	384	-	384	0.089	600 / 1.82
V-Co-L/16	918M	183	35	218	**0.095**	500 / **1.72**

results demonstrate that a carefully designed co-denoising recipe is both effective and scalable for representation-aligned pixel-space generation.

Computational efficiency. We further compare training GFLOPs and inference throughput in Table 6. Although V-Co introduces an additional DINOv2 encoder during training, this cost is largely offset by the improved parameter efficiency of the co-denoising backbone. Specifically, V-Co-B achieves nearly identical total training GFLOPs to JiT-L (88 vs. 89), while matching its generation quality (FID 2.35 vs. 2.36) and providing higher inference throughput (0.484 vs. 0.329 images/s). V-Co-L improves over JiT-H/G in FID (1.72 vs. 1.86/1.82), while requiring substantially fewer total training GFLOPs than JiT-G (218 vs. 384) and slightly exceeding its inference throughput (0.095 vs. 0.089 images/s). These results suggest that visual co-denoising improves not only generation quality, but also the quality–compute trade-off of pixel-space diffusion models.

5 Conclusion

We presented V-Co, a visual co-denoising framework for representation-aligned pixel-space generation. We identify four key ingredients for effective co-denoising: a fully dual-stream backbone, structural semantic-to-pixel masking for classifier-free guidance, a perceptual-drifting hybrid loss, and RMS-based feature rescaling. Together, they form a simple and scalable recipe that improves semantic alignment and generative quality, with strong ImageNet results and clear scalability with model size and training duration. We hope this work can inspire future research on co-denoising and representation-aligned generative modeling.

Acknowledgment

This work was supported by ONR Grant N00014-23-1-2356, ARO Award W911NF2110220, DARPA ECOLE Program No. HR00112390060, NSF-AI Engage Institute DRL-2112635 and NVIDIA Academic Grant Program. The views contained in this article are those of the authors and not of the funding agency.

References

1. Baade, A., Chan, E.R., Sargent, K., Chen, C., Johnson, J., Adeli, E., Fei-Fei, L.: Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. arXiv preprint arXiv:2602.11401 (2026)
2. Bi, H., Tan, H., Xie, S., Wang, Z., Huang, S., Liu, H., Zhao, R., Feng, Y., Xiang, C., Rong, Y., et al.: Motus: A unified latent action world model. arXiv preprint arXiv:2512.13030 (2025)
3. Chang, H., Cha, B., Ye, J.C.: Dino-sae: Dino spherical autoencoder for high-fidelity image reconstruction and generation. arXiv preprint arXiv:2601.22904 (2026)
4. Chefer, H., Singer, U., Zohar, A., Kirstain, Y., Polyak, A., Taigman, Y., Wolf, L., Sheynin, S.: Videojam: Joint appearance-motion representations for enhanced motion generation in video models. In: Forty-second International Conference on Machine Learning (2025)
5. Chen, S., Ge, C., Zhang, S., Sun, P., Luo, P.: Pixelflow: Pixel-space generative models with flow. arXiv preprint arXiv:2504.07963 (2025)
6. Chen, Z., Zhu, J., Chen, X., Zhang, J., Hu, X., Zhao, H., Wang, C., Yang, J., Tai, Y.: Dip: Taming diffusion models in pixel space. arXiv preprint arXiv:2511.18822 (2025)
7. Crowson, K., Baumann, S.A., Birch, A., Abraham, T.M., Kaplan, D.Z., Shippole, E.: Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In: Forty-first International Conference on Machine Learning (2024)
8. Dang, L., Li, Z., Li, J., Zhang, H., An, L., Liu, Y., Wu, Q.: Syncmv4d: Synchronized multi-view joint diffusion of appearance and motion for hand-object interaction synthesis. arXiv preprint arXiv:2511.19319 (2025)
9. Dang, L., Shao, R., Zhang, H., MIN, W., Liu, Y., Wu, Q.: Svimo: Synchronized diffusion for video and motion generation in hand-object interaction scenarios. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Deng, M., Li, H., Li, T., Du, Y., He, K.: Generative modeling via drifting. arXiv preprint arXiv:2602.04770 (2026)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
13. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
14. Han, X., Emad, Y., Hall, M., Nguyen, J., Padthe, K., Robbins, L., Bar, A., Chen, D., Drozdzal, M., Elbayad, M., et al.: Tv2tv: A unified framework for interleaved language and video generation. arXiv preprint arXiv:2512.05103 (2025)
15. Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: International Conference on Machine Learning. pp. 13213–13232. PMLR (2023)
16. Hoogeboom, E., Mensink, T., Heek, J., Lamerigts, K., Gao, R., Salimans, T.: Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18062–18071 (2025)

17. Hu, Z., Lai, C.H., Wu, G., Mitsufuji, Y., Ermon, S.: Meanflow transformers with representation autoencoders. arXiv preprint arXiv:2511.13019 (2025)
18. Huang, J., Zhang, Y., He, X., Gao, Y., Cen, Z., Xia, B., Zhou, Y., Tao, X., Wan, P., Jia, J.: Unityvideo: Unified multi-modal multi-task learning for enhancing world-aware video generation. arXiv preprint arXiv:2512.07831 (2025)
19. Jabri, A., Fleet, D.J., Chen, T.: Scalable adaptive computation for iterative generation. In: International Conference on Machine Learning. pp. 14569–14589. PMLR (2023)
20. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
22. Kouzelis, T., Karypidis, E., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: Boosting generative image modeling via joint image-feature synthesis. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
23. Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., Zheng, L.: Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18262–18272 (2025)
24. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
25. Li, T., Sun, Q., Fan, L., He, K.: Fractal generative models. arXiv preprint arXiv:2502.17437 (2025)
26. Lu, Y., Lu, S., Sun, Q., Zhao, H., Jiang, Z., Wang, X., Li, T., Geng, Z., He, K.: One-step latent-free image generation with pixel mean flows. arXiv preprint arXiv:2601.22158 (2026)
27. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: European Conference on Computer Vision. pp. 23–40. Springer (2024)
28. Ma, Z., Wei, L., Wang, S., Zhang, S., Tian, Q.: Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. arXiv preprint arXiv:2511.19365 (2025)
29. Ma, Z., Xu, R., Zhang, S.: Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. arXiv preprint arXiv:2602.02493 (2026)
30. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
31. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022)
33. Singh, J., Leng, X., Wu, Z., Zheng, L., Zhang, R., Shechtman, E., Xie, S.: What Matters for Representation Alignment: Global Information or Spatial Structure? arXiv preprint arXiv:2512.10794 (2025)
34. Tian, Y., Chen, H., Zheng, M., Liang, Y., Xu, C., Wang, Y.: U-repa: Aligning diffusion u-nets to vits. arXiv preprint arXiv:2503.18414 (2025)
35. Tong, S., Zheng, B., Wang, Z., Tang, B., Ma, N., Brown, E., Yang, J., Fergus, R., LeCun, Y., Xie, S.: Scaling text-to-image diffusion transformers with representation autoencoders. arXiv preprint arXiv:2601.16208 (2026)

36. Tschannen, M., Pinto, A.S., Kolesnikov, A.: Jetformer: An autoregressive generative model of raw images and text. In: The Thirteenth International Conference on Learning Representations (2025)
37. Wang, M., Jiang, D., Li, L., Lin, Y., Shen, G., Kong, X., Liu, Y., Dai, G., Wang, J.: Vae-repa: Variational autoencoder representation alignment for efficient diffusion training. arXiv preprint arXiv:2601.17830 (2026)
38. Wang, S., Gao, Z., Zhu, C., Huang, W., Wang, L.: Pixnerd: Pixel neural field diffusion. arXiv preprint arXiv:2507.23268 (2025)
39. Wang, S., Tian, Z., Huang, W., Wang, L.: Ddt: Decoupled diffusion transformer. arXiv preprint arXiv:2504.05741 (2025)
40. Wang, Z., Zhao, W., Zhou, Y., Li, Z., Liang, Z., Shi, M., Zhao, X., Zhou, P., Zhang, K., Wang, Z., et al.: Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. arXiv preprint arXiv:2505.16792 (2025)
41. Wu, J., Lian, H., Hao, D., Tian, Y., Shi, Q., Chen, B., Jiang, H., Tong, Y.: Does hearing help seeing? investigating audio-video joint denoising for video generation. arXiv preprint arXiv:2512.02457 (2025)
42. Yang, L., Qi, L., Li, X., Li, S., Jampani, V., Yang, M.H.: Unified dense prediction of video diffusion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28963–28973 (2025)
43. Yang, Y., Sheng, H., Cai, S., Lin, J., Wang, J., Deng, B., Lu, J., Wang, H., Ye, J.: Echomotion: Unified human video and motion generation via dual-modality diffusion transformer. arXiv preprint arXiv:2512.18814 (2025)
44. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
45. Yu, Y., Xiong, W., Nie, W., Sheng, Y., Liu, S., Luo, J.: Pixeldit: Pixel diffusion transformers for image generation. arXiv preprint arXiv:2511.20645 (2025)
46. Zhang, X., Liao, J., Zhang, S., Meng, F., Wan, X., Yan, J., Cheng, Y.: Videorepa: Learning physics for video generation through relational alignment with foundation models. arXiv preprint arXiv:2505.23656 (2025)
47. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025)
48. Zheng, R., Wang, J., Reed, S., Bjorck, J., Fang, Y., Hu, F., Jang, J., Kundalia, K., Lin, Z., Magne, L., et al.: Flare: Robot learning with implicit world modeling. arXiv preprint arXiv:2505.15659 (2025)
49. Zhong, Z., Yan, H., Li, J., Liu, X., Gong, X., Zhang, T., Song, W., Chen, J., Zheng, X., Wang, H., et al.: Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. arXiv preprint arXiv:2508.18269 (2025)