EXTRA DIMENSIONS, SIMPLER FLOWS: ACCELERATING FLOW MATCHING TRAINING WITH PRETRAINED VISUAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

It is an effective strategy to accelerate the training of diffusion and flow-based generative models by jointly modeling image latent features and semantic representations extracted from pretrained vision encoders. However, stronger encoders do not always improve generation quality and can hinder optimization. We show that a semantic representation reduces velocity ambiguity only when it provides velocity-relevant information that is unavailable from the noisy image state, thereby distinguishing otherwise ambiguous velocity targets. Motivated by this insight, we propose Joint Semantic Flow Matching (JsFlow), which jointly transports the image state and semantic variables from multiple pretrained encoders. JsFlow accelerates semantic transport to provide more informative semantic states throughout the trajectory. Joint image–semantic prediction can still introduce gradient interference through the shared backbone. To address this issue, we predict semantic velocities through a separate branch that takes detached backbone features as input. This preserves semantic guidance during transport while preventing semantic-loss gradients from updating the image backbone. On ImageNet 256×256, JsFlow with LightningDiT-XL/1 achieves an unguided FID of 5.91 after only 50K iterations. It outperforms SiT-XL/2 trained for 7M iterations (FID 8.30) and LightningDiT-XL/1 trained for 4M iterations (FID 6.47), requiring only $1/140$ and $1/80$ of their training iterations, respectively.

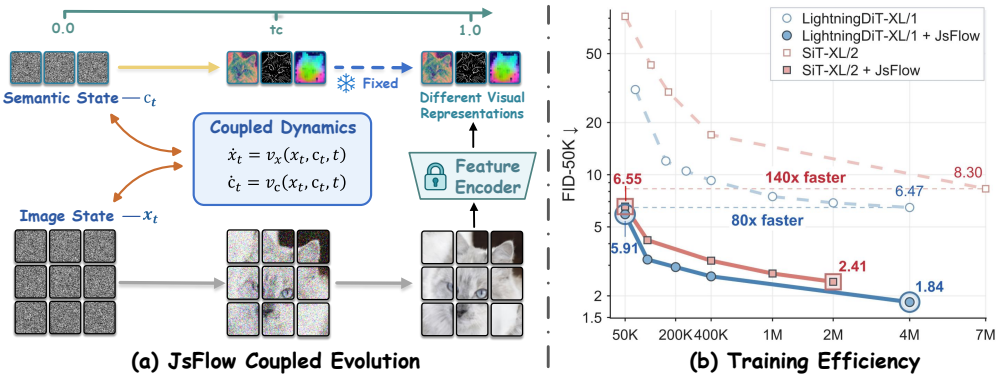


Figure 1: **(a) JsFlow Coupled Evolution.** JsFlow jointly evolves the image state x_t and multi-view semantic state c_t through coupled dynamics. The semantic state evolves towards high-level class tokens extracted from multiple pretrained encoders, reaching fixed semantic anchors at $t_c < 1$, while the image state continues evolving. **(b) Training Efficiency.** JsFlow uses approximately $80\times$ and $140\times$ fewer training iterations than LightningDiT-XL/1 and SiT-XL/2, respectively.

1 INTRODUCTION

Flow Matching (FM) learns continuous velocity fields that transport samples from a simple prior distribution to the data distribution along continuous probability paths, providing a flexible and ef-

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.

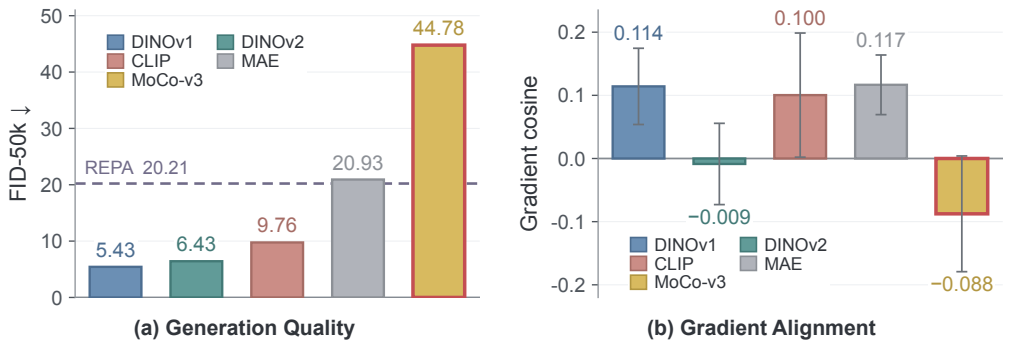


Figure 2: **Joint image–semantic modeling with different visual representations. (a) Generation Quality.** Different visual representations yield different generation performance. MoCo-v3 features substantially degrade FID-50K compared with the REPA baseline. **(b) Gradient Alignment.** Cosine similarity between the semantic-loss and image-loss gradients on the shared backbone. Negative values indicate conflicting update directions.

fective framework for high-quality image generation. (Lipman et al., 2023; Liu et al., 2022; Albergo & Vanden-Eijnden, 2023; Ma et al., 2024). However, it still requires substantial training computation to achieve high quality generation(Peebles & Xie, 2023; Esser et al., 2024). Recent methods improve training efficiency through pretrained visual representations, using feature alignment (Yu et al., 2025; Leng et al., 2025), semantic conditioning (Li et al., 2024), or enriched latent spaces (Yao et al., 2025; Zheng et al., 2025). In particular, jointly modeling latent image and high-level semantics has emerged as an effective approach to accelerating convergence and improving generation quality (Wu et al., 2025; Kouzelis et al., 2025; Pan et al., 2025).

However, strong visual representations do not uniformly benefit generation. Generation quality varies substantially with the choice of pretrained visual encoder used to extract the global class token for the evolving semantic representation (see Fig. 2(a)), and some exhibit conflicting semantic and image gradients on the shared backbone (see Fig. 2(b)). Such interference is also recognized in multi-task optimization (Yu et al., 2020; Liu et al., 2021). These observations raise a central question: ***What kind of semantic representation simplifies generative learning?***

We study this problem through velocity ambiguity. Different noise–data pairs may share an intermediate state but have different target velocities, creating a one-to-many mapping whose conditional variance is irreducible under the squared loss. Prior work reduces this ambiguity through improved couplings or lower target variance (Pooladian et al., 2023; Yang et al., 2026). We prove that semantic augmentation cannot increase this variance and reduces it when the semantic representations distinguish ambiguous velocity targets.

Inspired by this insight, we propose Joint Semantic Flow Matching (JsFlow). JsFlow augments the image state with multiple semantic variables derived from global class tokens extracted by CLIP, DINOv1, and DINOv2 (Radford et al., 2021; Caron et al., 2021; Oquab et al., 2024). It jointly transports the image and semantic variables, with semantic transport finishing earlier at $t_c < 1$. The semantic endpoints are then held fixed as anchors for the remaining image transport. We further introduce an enhanced variant, JsFlow†, which predicts semantic velocities through a dedicated branch. This design prevents semantic supervision from updating the image backbone.

Our main contributions are summarized as follows:

- We prove that semantic representations reduce velocity ambiguity when they provide information beyond the image state, explaining when visual representations benefit FM.

- We propose JsFlow, which jointly transports images and multiple semantic representations, with semantics completed early to guide the remaining image transport. We further introduce JsFlow† to decouple semantic supervision from the image backbone.

• JsFlow consistently accelerates convergence across model scales and backbones. On ImageNet 256×256, JsFlow achieves an unguided FID of **5.91** after **50K** iterations. JsFlow† further improves the FID to **5.44**. After **4M** iterations, JsFlow reaches FIDs of **1.84** without guidance and **1.18** with guidance.

2 Related Work

2.1 Semantic Representation-Enhanced Generative Models

Recent works have increasingly incorporated pretrained visual representations into diffusion and flow-based generative models. RAE (Zheng et al., 2025) and related approaches replace or enrich conventional VAE latents with semantically stronger visual representations, while REPA (Yu et al., 2025) and REPA-E (Leng et al., 2025) improve generative representation learning through feature alignment. Beyond representation alignment, joint modeling of low-level image latents and high-level semantics has also been widely explored. REG (Wu et al., 2025) and ReDi (Kouzelis et al., 2025) explicitly incorporate semantic features into the generative state, while SFD (Pan et al., 2025) further studies coordinated semantic–texture modeling within a unified generative framework, with specialized schedules to regulate their generation processes. These methods use semantic information to accelerate convergence and improve generation quality. JsFlow provides a theoretical perspective on these benefits through velocity ambiguity.

2.2 Variance Reduction in Diffusion and Flow Models

Variance reduction has been widely explored to stabilize diffusion and flow model training. Existing diffusion methods use timestep importance sampling or control variates to reduce gradient noise (Nichol & Dhariwal, 2021; Jeha et al., 2024), while Stable Target Field and nearest-neighbor estimators aggregate multiple conditional scores into more stable targets (Xu et al., 2023; Niedoba et al., 2024). In Flow Matching, OT-CFM and Multisample Flow Matching redesign source–target couplings (Tong et al., 2024; Pooladian et al., 2023), stream-level Flow Matching models multiple stochastic paths (Wei & Ma, 2025), and Stable Velocity aggregates multiple reference velocities (Yang et al., 2026). Complementary semantic representations can reduce ambiguity in image-velocity targets and provide more stable estimates.

3 Method

In this section, we develop JsFlow by extending FM to joint image–semantic transport. Section 3.1 reviews FM through stochastic interpolants. Section 3.2 defines joint FM in an augmented state space. Section 3.3 shows how semantic augmentation reduces velocity ambiguity. Based on this analysis, Section 3.4 presents the main designs of JsFlow.

3.1 Preliminaries

We present a brief overview of flow and diffusion-based models through the unified perspective of stochastic interpolants.

Flow Matching. FM learns a continuous transport from a simple prior distribution to the data distribution through a time-dependent velocity field. Let $\mathbf{x}_1 \sim p_{\text{data}}$ denote a data sample and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denote Gaussian noise. Following the stochastic-interpolant formulation, we define a probability path

$$\mathbf{x}_t = \alpha_t \mathbf{x}_1 + \sigma_t \boldsymbol{\epsilon}, \qquad t \in [0, 1], \tag{1}$$

where α_t and σ_t smoothly interpolate between noise and data, satisfying $\alpha_0 = 0$, $\sigma_0 = 1$, $\alpha_1 = 1$, and $\sigma_1 = 0$. The corresponding conditional velocity is

$$\mathbf{u}_t = \frac{d\mathbf{x}_t}{dt} = \dot{\alpha}_t \mathbf{x}_1 + \dot{\sigma}_t \boldsymbol{\epsilon}. \tag{2}$$

FM trains a neural velocity field $\mathbf{v}_\theta(\mathbf{x}_t, t)$ to predict this conditional velocity by minimizing

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\mathbf{x}_1, \boldsymbol{\epsilon}, t} \left[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{u}_t\|_2^2 \right]. \tag{3}$$

Under this squared regression objective, the population-optimal velocity field is given by the conditional expectation

$$\mathbf{v}(\mathbf{x}, t) = \mathbb{E}\left[\mathbf{u}_t \mid \mathbf{x}_t = \mathbf{x}\right]. \tag{4}$$

Thus, $\mathbf{v}(\mathbf{x}, t)$ averages the velocities of all conditional transport paths that reach the same intermediate state $\mathbf{x}$ at time t. At inference time, samples are generated by integrating the learned ODE $\dot{\mathbf{x}}_t = \mathbf{v}_\theta(\mathbf{x}_t, t)$ from the prior distribution at $t = 0$ to the data distribution at $t = 1$.

3.2 Joint Flow Matching in an Augmented State Space

Joint image–semantic transport. Given an image, let $\mathbf{X}_1$ denote its VAE latent and $\mathbf{C}_1$ its semantic representation, with independent Gaussian sources $\mathbf{X}_0, \mathbf{C}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We lift Flow Matching to the augmented state $\mathbf{Z}_t = [\mathbf{X}_t; \mathbf{C}_t]$ by defining

$$\mathbf{X}_t = (1 - \alpha(t))\mathbf{X}_0 + \alpha(t)\mathbf{X}_1, \qquad \mathbf{C}_t = (1 - \beta(t))\mathbf{C}_0 + \beta(t)\mathbf{C}_1, \tag{5}$$

where $\alpha(t)$ and $\beta(t)$ independently control the generation progress of image and semantic coordinates, satisfying $\alpha(0) = \beta(0) = 0$ and $\alpha(1) = \beta(1) = 1$.

The corresponding conditional velocity is

$$\mathbf{U}_Z(t) := \begin{bmatrix} \mathbf{U}_X(t) \\ \mathbf{U}_C(t) \end{bmatrix} = \begin{bmatrix} \dot{\alpha}(t)(\mathbf{X}_1 - \mathbf{X}_0) \\ \dot{\beta}(t)(\mathbf{C}_1 - \mathbf{C}_0) \end{bmatrix}. \tag{6}$$

JsFlow models this joint field with a shared network, $\mathbf{v}_\theta^Z(\mathbf{Z}_t, t) = [\mathbf{v}_\theta^X(\mathbf{Z}_t, t); \mathbf{v}_\theta^C(\mathbf{Z}_t, t)]$, and optimizes

$$\mathcal{L} = \underbrace{\mathbb{E}\left[\left\|\mathbf{v}_\theta^X(\mathbf{Z}_t, t) - \mathbf{U}_X(t)\right\|_2^2\right]}_{\mathcal{L}_{\text{img}}\ \text{(image loss)}} + \lambda_C \underbrace{\mathbb{E}\left[\left\|\mathbf{v}_\theta^C(\mathbf{Z}_t, t) - \mathbf{U}_C(t)\right\|_2^2\right]}_{\mathcal{L}_C\ \text{(semantic loss)}}. \tag{7}$$

Here, λ_C controls the strength of semantic supervision. Since our primary goal is image generation, the following analysis focuses on the image branch. Importantly, semantics are not merely auxiliary targets: they form jointly transported coordinates of the augmented generative state.

3.3 Semantic Augmentation Reduces Velocity Ambiguity

We next characterize when augmenting the image state with semantics can simplify velocity prediction. Under the squared Flow Matching objective, uncertainty in the conditional velocity target appears as an irreducible component of the regression risk.

Intrinsic velocity ambiguity. Let $\mathcal{D}$ index a random training run, $\widehat{\mathbf{v}}_{\mathcal{D}}(\mathbf{X}_t, t)$ denote its velocity predictor, and $\overline{\mathbf{v}}(\mathbf{X}_t, t) := \mathbb{E}_{\mathcal{D}}[\widehat{\mathbf{v}}_{\mathcal{D}}(\mathbf{X}_t, t)]$ denote the mean prediction across training runs.

Theorem 1 (Bias–variance decomposition of Flow Matching). For squared FM regression,

$$\begin{aligned} &\mathbb{E}_{\mathcal{D},\, \mathbf{U}_X(t)|\mathbf{X}_t,t}\left[\|\widehat{\mathbf{v}}_{\mathcal{D}}(\mathbf{X}_t, t) - \mathbf{U}_X(t)\|_2^2 \mid \mathbf{X}_t, t\right] \\ &= \underbrace{\|\overline{\mathbf{v}}(\mathbf{X}_t, t) - \mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{X}_t, t]\|_2^2}_{\text{squared bias}} + \underbrace{\operatorname{tr}\operatorname{Cov}_{\mathcal{D}}(\widehat{\mathbf{v}}_{\mathcal{D}}(\mathbf{X}_t, t))}_{\text{estimation variance}} + \underbrace{\operatorname{tr}\operatorname{Cov}(\mathbf{U}_X(t) \mid \mathbf{X}_t, t)}_{\text{irreducible velocity variance}}. \end{aligned} \tag{8}$$

The proof is deferred to Appendix B.1.

Thus, the population-optimal predictor is $\mathbf{v}_X^\star = \mathbb{E}[\mathbf{U}_{\mathbf{X}}(t) \mid \mathbf{X}_t, t]$, and its minimum achievable risk is

$$\mathcal{L}_{\text{FM}}^\star(t) = \mathbb{E}\left[\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{X}_t, t\right)\right]. \tag{9}$$

We refer to this irreducible uncertainty as *velocity ambiguity*. It creates a one-to-many mapping from an observed state to its target velocities, making the velocity field harder for a neural network to learn. This ambiguity cannot be eliminated by optimization alone.

Reducing ambiguity through semantic augmentation. The same decomposition applies after adding semantic variables to the state. We define the irreducible variances conditioned on the image-only and augmented states by

$$\mathcal{V}_X(t) = \mathbb{E}\left[\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{X}_t, t\right)\right], \quad \mathcal{V}_Z(t) = \mathbb{E}\left[\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right)\right]. \tag{10}$$

Theorem 2 (Semantic augmentation reduces velocity ambiguity). The reduction in the minimum achievable image-velocity regression risk is

$$\Delta\mathcal{L}^{\star}_{\text{img}}(t) := \mathcal{V}_X(t) - \mathcal{V}_Z(t) = \mathbb{E}\left[\left\|\mathbb{E}\left[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right] - \mathbb{E}\left[\mathbf{U}_X(t) \mid \mathbf{X}_t, t\right]\right\|_2^2\right] \geq 0. \tag{11}$$

The proof is provided in Appendix B.1.

Theorem 2 shows that semantic augmentation never increases the irreducible image-velocity variance. It reduces the variance only when $\mathbf{C}_t$ provides velocity information not contained in $\mathbf{X}_t$. No reduction occurs when $\mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t] = \mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{X}_t, t]$ almost surely. The variance becomes zero if $\mathbf{Z}_t$ fully determines $\mathbf{U}_X(t)$. Thus, a useful semantic coordinate only needs to resolve the velocity uncertainty left by $\mathbf{X}_t$. Theorem 2 also directly yields the following corollary.

Corollary 1 (Monotonic reduction with complementary semantics). Let

$$\mathbf{Z}_t^{(k)} = \left[\mathbf{X}_t^\top, \mathbf{C}_t^{(1)\top}, \ldots, \mathbf{C}_t^{(k)\top}\right]^\top. \tag{12}$$

Repeated application of Theorem 2 yields

$$\mathcal{V}_X(t) \geq \mathcal{V}_{Z^{(1)}}(t) \geq \mathcal{V}_{Z^{(2)}}(t) \geq \cdots \geq \mathcal{V}_{Z^{(k)}}(t) \geq 0. \tag{13}$$

Thus, each additional semantic coordinate can further reduce velocity ambiguity when it contributes information not already contained in the current augmented state. This motivates combining complementary, rather than redundant, semantic representations.

3.4 Semantic-Guided Velocity Simplification

To reduce the irreducible velocity variance, standard JsFlow combines multi-view semantic augmentation with asynchronous transport. We further enhance JsFlow with gradient decoupling and denote this variant by $\text{JsFlow}^\dagger$.

Multi-view semantic augmentation. We augment the image state with multiple semantic coordinates:

$$\mathbf{Z}_t = \left[\mathbf{X}_t^\top, \mathbf{C}_t^{(1)\top}, \ldots, \mathbf{C}_t^{(K)\top}\right]^\top. \tag{14}$$

Motivated by Corollary 1, additional views can reduce irreducible velocity ambiguity when they provide information not already captured by the current state. We use global class tokens from CLIP, DINOv2, and DINOv1 as complementary semantic representations.

Semantic-first asynchronous transport. We let semantic transport finish before image transport by setting $\beta(t) = 1$ for $t \geq t_c$, where $t_c < 1$. Consequently, for $t \geq t_c$,

$$\mathbf{C}_t = \mathbf{C}_1, \quad \dot{\mathbf{C}}_t = \mathbf{0}. \tag{15}$$

Accelerating semantic transport makes richer semantic information available at each intermediate time. This design strengthens semantic conditioning throughout image transport and helps reduce image velocity ambiguity across the entire trajectory.

Gradient-decoupled semantic prediction. To prevent semantic supervision from directly interfering with image optimization, $\text{JsFlow}^\dagger$ predicts semantic velocities through a dedicated branch. The semantic branch takes features from the image backbone as input. We detach these features before passing them to the branch, so the semantic loss does not update the image backbone. Considering this branch introduces additional parameters and computation, we include it only in the enhanced variant.

Representation alignment. Both standard JsFlow and $\text{JsFlow}^\dagger$ use REPA to align intermediate image features with DINOv2 patch features extracted from clean images. This patch-level supervision complements the global semantic tokens used for joint transport.

Training objective. We train both JsFlow and $\text{JsFlow}^\dagger$ using the following objective:

$$\mathcal{L}_{\text{JsFlow}} = \mathcal{L}_{\text{img}} + \lambda_C \mathcal{L}_C + \lambda_{\text{REPA}} \mathcal{L}_{\text{REPA}}, \tag{16}$$

where $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_C$ supervise the image and semantic velocities, respectively, and $\mathcal{L}_{\text{REPA}}$ provides representation-level supervision.

Implementation details, comparisons with related methods, and analyses of how the individual designs affect velocity variance are provided in Appendix C.

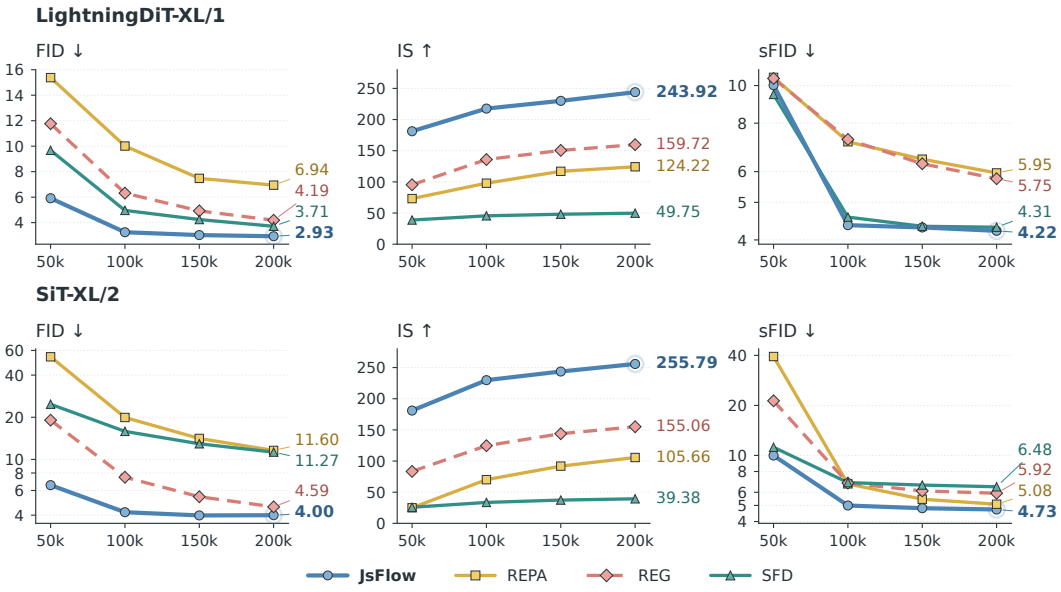


Figure 3: **Training convergence comparison on ImageNet** 256×256**.** We compare JsFlow with representative representation-enhanced baselines on LightningDiT-XL/1 and SiT-XL/2. All models are trained with a learning rate of 2×10^{-4}. JsFlow consistently achieves better FID, sFID, and IS, demonstrating faster convergence across both backbones.

4 Experiments

We validate the performance of JsFlow and analyze the effect of its key designs through extensive experiments. In particular, we investigate the following questions:

Q1 Can JsFlow significantly improve generation performance? (Section 4.2)

Q2 What are the contributions of the proposed components in JsFlow? (Section 4.3)

Q3 Can the empirical gains of JsFlow be explained from the perspective of velocity ambiguity reduction? (Section 4.4)

4.1 Setup

Implementation details. We conduct class-conditional generation experiments on ImageNet (Deng et al., 2009) at 256×256 resolution using LightningDiT-XL/1 (Yao et al., 2025) and SiT-XL/2 (Ma et al., 2024). LightningDiT uses VA-VAE to produce $16 \times 16 \times 32$ image latents, whereas SiT uses SD-VAE (Rombach et al., 2022) to produce $32 \times 32 \times 4$ latents. Unless otherwise specified, JsFlow uses 3 semantic tokens from frozen CLIP ViT-B, DINOv1 ViT-B, and DINOv2 ViT-B encoders (Radford et al., 2021; Caron et al., 2021; Oquab et al., 2024). We train all models with AdamW (Loshchilov & Hutter, 2019) and a global batch size of 256. We set $\lambda_C = 0.03$, $\lambda_{\text{REPA}} = 0.5$, and $t_c = 0.7$. Additional implementation details are provided in the Appendix.

Sampling. We generate samples from EMA model weights using an Euler Maruyama sampler (Ma et al., 2024) with 250 sampling steps unless otherwise stated. For evaluations with classifier-free guidance (CFG) (Ho & Salimans, 2022), we apply class-label guidance with image CFG $w_1 = 3.0$ and semantic CFG $w_2 = 1.25$.

Evaluation protocol. We report Fréchet Inception Distance (FID) (Heusel et al., 2017), spatial FID (sFID) (Nash et al., 2021), Inception Score (IS) (Salimans et al., 2016), and Precision (Prec.) and Recall (Rec.) (Kynkäänniemi et al., 2019). All metrics are computed on 50K generated images unless otherwise specified.

Table 1: **FID comparison on ImageNet** 256×256 **without guidance** across different model sizes and training iterations.

Model	#Params	Iter.	FID↓
DiT-B/2	130M	400K	42.89
LightningDiT-B/1	130M	400K	22.63
+ REPA	130M	400K	21.42
+ REG	132M	400K	11.04
+ JsFlow (Ours)	143M	400K	**7.25**
DiT-L/2	458M	400K	23.31
LightningDiT-L/1	458M	400K	10.11
+ REPA	458M	400K	7.46
+ REG	460M	400K	3.86
+ JsFlow (Ours)	474M	400K	**2.53**
DiT-XL/2	675M	400K	19.51
DiT-XL/2	675M	7M	9.58
LightningDiT-XL/1	675M	400K	9.26
LightningDiT-XL/1	675M	4M	6.47
+ REPA	675M	400K	6.49
+ REPA	675M	4M	5.86
+ REG	677M	400K	3.29
+ REG	677M	4M	2.20
+ JsFlow (Ours)	684M	50K	5.91
+ JsFlow† (Ours)	784M	50K	5.44
+ JsFlow (Ours)	684M	100K	3.24
+ JsFlow (Ours)	684M	400K	2.59
+ JsFlow† (Ours)	784M	400K	2.56
+ JsFlow (Ours)	684M	1M	2.51
+ JsFlow (Ours)	684M	4M	**1.84**

Table 2: **System-level comparison of class-conditional generation on ImageNet** 256×256 **with guidance.** Performance metrics are annotated with ↑ (higher is better) and ↓ (lower is better).

Model	Epochs	#Params	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Autoregressive Models							
VAR	350	2.0B	1.80	–	365.4	0.83	0.57
MAR	800	943M	1.55	–	303.7	0.81	0.62
RAR-XXL	400	1.5B	1.48	–	326.0	0.80	0.63
xAR-H	800	1.1B	1.24	–	301.6	0.83	0.64
Latent Diffusion and Flow Models							
DiT-XL/2	1400	675M	2.27	4.60	278.2	0.83	0.57
SiT-XL	1400	675M	2.06	4.50	270.3	0.82	0.59
U-ViT-H/2	400	501M	2.29	5.68	263.9	0.82	0.57
MaskDiT	1600	675M	2.28	5.67	276.6	0.80	0.61
DDT	400	675M	1.26	–	310.6	0.79	0.65
DiffiT	400	561M	1.73	4.54	276.5	0.80	0.62
DiMR	800	505M	1.70	–	289.0	0.79	0.63
Representation-Enhanced Generative Models							
REPA	800	675M	1.42	4.70	305.7	0.80	0.65
REG	800	677M	1.36	4.25	299.4	0.77	0.66
ReDi	800	675M	1.61	4.66	295.1	0.78	0.64
REPA-E	800	675M	1.12	4.09	302.9	0.79	0.66
FAE	800	675M	1.29	–	268.0	0.80	0.64
RiT	800	676M	1.14	–	299.7	0.80	0.63
SFD	800	675M	1.06	3.89	267.0	0.78	0.67
JsFlow (Ours)	80	684M	1.63	4.02	278.7	0.81	0.62
JsFlow (Ours)	800	684M	1.18	3.93	293.7	0.78	0.68



Figure 4: **Selected samples on ImageNet** 256×256 **from the LightningDiT-XL/1 + JsFlow model.** We train the model for 4M iterations and use classifier-free guidance with $w = 3.0$.

4.2 Generation Performance

Accelerating training convergence. As shown in Table 1, JsFlow achieves an unguided FID of 5.91 after only 50K iterations. It outperforms DiT-XL/2 (Peebles & Xie, 2023) at 7M iterations (9.58) and LightningDiT-XL/1 (Yao et al., 2025) at 4M iterations (6.47), using $140\times$ and $80\times$ fewer iterations, respectively. JsFlow† further improves the 50K FID to 5.44. With extended training, JsFlow reaches an FID of 1.84 at 4M iterations.

Consistent gains across scales and backbones. At 400K iterations, JsFlow achieves FIDs of 7.25, 2.53, and 2.59 on LightningDiT-B/1, L/1, and XL/1, outperforming the corresponding REPA and REG (Wu et al., 2025). Similar gains are observed on both LightningDiT and SiT backbones in Figure 3. At 200K iterations, JsFlow achieves FIDs of 2.93 on LightningDiT-XL/1 and 4.00 on SiT-XL/2 (Ma et al., 2024), together with the best IS and sFID among the compared methods.

Comparison with SOTA methods. With guidance, JsFlow achieves an FID of 1.63 after only 80 epochs, outperforming DiT-XL/2 (2.27) and SiT-XL (2.06) trained for 1400 epochs. At 800 epochs,

Table 3: **Generation quality and computational cost.** All methods use the LightningDiT-XL/1 backbone and are evaluated at 50K steps without guidance.

Method	#Params	FLOPs↓	Latency(s)↓	FID↓	IS↑
REPA	675M	237.3	3.30	15.38	73.2
REG	677M	238.2	3.56	11.76	95.4
JsFlow	684M	240.6	4.24	5.91	181.2
JsFlow†	784M	275.0	4.87	**5.44**	**200.4**

Table 4: **Comparison of feature sources at 50K training steps.** JsFlow† uses Wavelet or Fourier features. Time reports feature extraction latency in ms/image.

Method	Features	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	Time↓
REPA	–	15.38	10.50	73.2	0.722	0.538	–
REG	DINOv2	11.76	10.43	95.4	**0.751**	0.516	21.26
JsFlow†	Wavelet	**11.37**	7.92	**96.7**	0.750	0.550	**0.014**
JsFlow†	Fourier	11.55	**7.28**	93.8	0.736	**0.556**	0.018

JsFlow reaches an FID of 1.18, compared with 1.42 for REPA, 1.36 for REG, and 1.61 for ReDi (Kouzelis et al., 2025). Selected samples are shown in Figure 4.

Computational efficiency. Tables 3 and 4 report model computation and feature-extraction cost. Compared with REG, standard JsFlow adds only 1.0% to both parameter count and FLOPs (684M and 240.6), while reducing FID by 49.7% to 5.91 and increasing IS from 95.4 to 181.2. Its latency increases from 3.56 to 4.24 seconds. JsFlow† further reaches an FID of 5.44 and an IS of 200.4 using 784M parameters, 275.0 FLOPs, and 4.87 seconds. Wavelet and Fourier require only 0.014 and 0.018 ms per image, over 1000× faster than DINOv2 (21.26 ms), while achieving FIDs of 11.37 and 11.55 compared with REG's 11.76.

4.3 Ablation Studies

Effect of semantic representations and multi-view augmentation. Tables 5 and 6 reveal clear differences among semantic representations. DINOv1 achieves the best single-encoder FID, whereas DINOv2 gives higher IS. MAE and MoCo-v3 obtain higher recall but suffer from much worse FID and precision, indicating a trade-off between coverage and fidelity. Combining CLIP, DINOv1, and DINOv2 yields the best FID and IS for both standard JsFlow (4.89 and 172.5) and JsFlow† (4.22 and 197.88), supporting the effectiveness of multi-view augmentation in improving generation beyond individual semantic representations.

Effect of the transition time. The transition time t_c determines when semantic generation finishes and the semantic state becomes a fixed anchor for subsequent image generation. It therefore controls the balance between forming reliable semantic anchors and using them to guide image refinement. Figure 5 shows that an intermediate transition works best, achieving an FID of 8.47 at $t_c = 0.7$. An earlier transition provides insufficient time to generate reliable semantic anchors, whereas a later transition leaves less time to refine the image under fixed semantic guidance. The observed optimum is consistent with this trade-off and suggests that semantic generation should finish early enough to guide the remaining image transport.

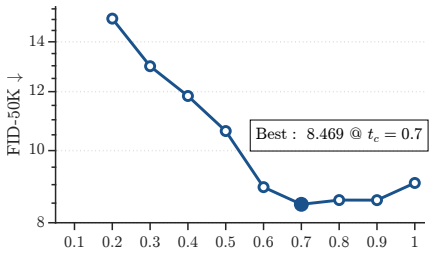


Figure 5: **Effect of the transition time.** JsFlow achieves the best generation quality when semantic transport finishes at $t_c = 0.7$.

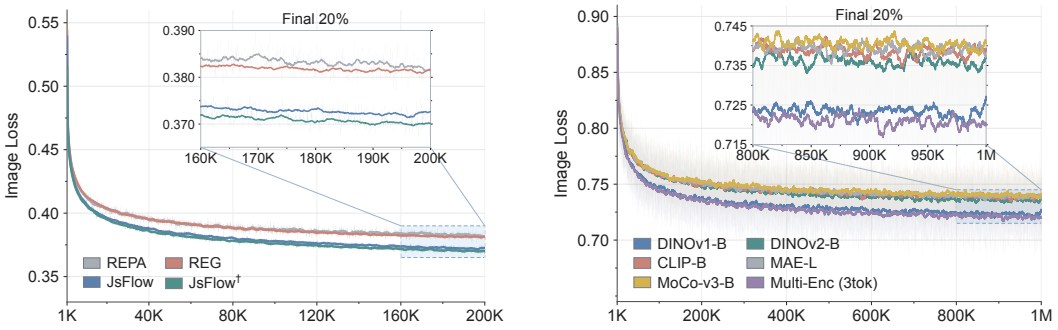
Effect of optimization decoupling. Tables 5 and 6 compare standard JsFlow with JsFlow† under the same setting. JsFlow† improves FID and IS for every semantic representation. With multi-view semantics, FID improves from 4.89 to 4.22 and IS increases from 172.5 to 197.88. The largest gain is observed with MoCo-v3, whose FID decreases from 44.78 to 18.22 and IS increases from 35.9 to 85.5. These results show that optimization decoupling improves generation across semantic representations, especially for weaker representations such as MoCo-v3.

Table 5: Semantic representation comparison with **JsFlow** on SiT-B/2 at 1M iterations without guidance.

Representation	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Multi-Enc	**4.89**	**5.71**	**172.5**	**0.81**	0.50
DINOv1 ViT-B	5.43	5.83	157.9	0.77	0.57
DINOv2 ViT-B	6.43	6.44	166.3	0.74	0.57
CLIP ViT-B	9.76	6.22	126.5	0.71	0.60
MAE ViT-L	20.93	8.32	81.1	0.60	0.63
MoCo-v3 ViT-B	44.78	7.75	35.9	0.46	**0.68**

Table 6: Semantic representation comparison with **JsFlow**† on SiT-B/2 at 1M iterations without guidance.

Representation	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Multi-Enc	**4.22**	6.06	**197.88**	**0.80**	0.53
DINOv1 ViT-B	4.31	**5.48**	174.09	0.78	0.57
DINOv2 ViT-B	5.02	5.92	180.73	0.76	0.59
CLIP ViT-B	7.38	5.87	148.71	0.73	0.59
MAE ViT-L	19.40	8.45	83.39	0.64	**0.63**
MoCo-v3 ViT-B	18.22	6.19	85.5	0.63	**0.63**



(a) Comparison across methods.

(b) Comparison across semantic encoders.

Figure 6: Image-branch regression loss across methods and semantic encoders. Lower MSE is generally associated with better generation quality, consistent with the FID results.

4.4 Velocity Ambiguity Analysis

JsFlow reduces velocity ambiguity by augmenting the image state with semantic information. Since conditional velocity variance is difficult to estimate directly, we use the late stage image branch MSE as an approximate measure. Under squared loss, the minimum MSE equals the expected conditional target variance once the predictor approaches its optimum. Lower variance also reduces target-induced gradient noise, leading to more stable and faster optimization.

Figure 6(a) shows that JsFlow and JsFlow† reduce the regression loss more rapidly and maintain lower late-stage MSE than REPA and REG. Figure 6(b) further shows that DINOv1 and the multi-encoder combination achieve the lowest losses among the tested semantic representations. These results suggest that semantic augmentation provides a clearer learning signal for image-velocity prediction. The loss trends also agree with the FID results, supporting the view that reducing velocity ambiguity facilitates optimization and improves generation quality.

5 Conclusion

In this work, we study how semantic representations facilitate FM through the lens of velocity ambiguity. We show that semantic augmentation cannot increase irreducible image-velocity variance and reduces it when the added representations provide complementary velocity-relevant information. Motivated by this insight, we propose JsFlow, which jointly transports image states and multi-view semantic variables. Accelerated semantic transport provides informative guidance throughout generation, while a dedicated semantic branch with detached backbone features reduces gradient interference. Experiments across model scales and backbones show faster convergence and better generation quality. Lower late-stage regression losses further support the connection between reduced conditional velocity variance and these gains. Overall, additional dimensions can simplify FM training when they contain complementary velocity-relevant information unavailable from the original state.

AI USE STATEMENT

We used generative AI tools to assist with manuscript translation, grammar checking, data preprocessing, and code writing. The authors reviewed the AI-assisted text and code for accuracy and correctness and take full responsibility for the final manuscript and implementation.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. 2023. URL https://arxiv.org/abs/2209.15571.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23164–23173, October 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020. URL https://arxiv.org/abs/2006.11239.

Paul Jeha, Will Grathwohl, Michael Riis Andersen, Carl Henrik Ek, and Jes Frellsen. Variance reduction of diffusion model's gradients with taylor approximation-based control variate. 2024. URL https://arxiv.org/abs/2408.12270.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14162–14171, June 2024.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi,

and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 16685–16714. Curran Associates, Inc., 2025. doi: 10.52202/085713-0563. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/186a213d720568b31f9b59c085a23e5a-Paper-Conference.pdf.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272, October 2025.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 125441–125468. Curran Associates, Inc., 2024. doi: 10.52202/079017-3985. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e304d374c85e385eb217ed4a025b6b63-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d27fdf2477ffbff837d73ef7ae23db9-Paper.pdf.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. 2022. URL https://arxiv.org/abs/2209.03003.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019. URL https://arxiv.org/abs/1711.05101.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. 2024. URL https://arxiv.org/abs/2401.08740.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7958–7968. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nash21a.html.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Matthew Niedoba, Dylan Green, Saeid Naderiparizi, Vasileios Lioutas, Jonathan Wilder Lavington, Xiaoxuan Liang, Yunpeng Liu, Ke Zhang, Setareh Dabiri, Adam Scibior, Berend Zwartsenberg, and Frank Wood. Nearest neighbour score estimators for diffusion generative models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38117–38144. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/niedoba24a.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. 2024. URL https://arxiv.org/abs/2304.07193.

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. 2025. URL https://arxiv.org/abs/2512.04926.

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

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28100–28127. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/pooladian23a.html.

Xiaozhen Qiao, Jingkai Zhao, Yuqiu Jiang, Xianda Guo, Zhe Sun, Hongyuan Zhang, and Xuelong Li. Class-aware prototype learning with negative contrast for test-time adaptation of vision-language models. 2025. URL https://arxiv.org/abs/2510.19802.

Xiaozhen Qiao, Peng Huang, Jiakang Yuan, Xianda Guo, Bowen Ye, Chaocan Xue, Ye Zheng, Zhe Sun, and Xuelong Li. Bidirectional prototype-reward co-evolution for test-time adaptation of vision-language models. *IEEE Transactions on Multimedia*, pp. 1–11, 01 2026a. doi: 10.1109/TMM.2026.3694557.

Xiaozhen Qiao, Da Zhang, Yubin Guo, Junyu Gao, Zhiyuan Zhao, and Xuelong Li. Semantic-aware temporal adaptation for uav anti-uav tracking. 2026b. URL https://arxiv.org/abs/2607.26511.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. 2016. URL https://arxiv.org/abs/1606.03498.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14274–14289. Curran Associates, Inc., 2022. doi: 10.52202/068431-1038. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/5bf2b802e24106064dc547ae9283bb0c-Paper-Conference.pdf.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. 2021. URL https://arxiv.org/abs/2011.13456.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1185–1194, October 2025.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. 2024. URL https://arxiv.org/abs/2302.00482.

Ganchao Wei and Li Ma. Stream-level flow matching with Gaussian processes. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66137–66154. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wei25i.html.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. 2025. URL https://arxiv.org/abs/2507.01467.

Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models. 2023. URL https://arxiv.org/abs/2302.00670.

Donglin Yang, Yongxing Zhang, Xin Yu, Liang Hou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Renjie Liao. Stable velocity: A variance perspective on flow matching. 2026. URL https://arxiv.org/abs/2602.05435.

Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56166–56189. Curran Associates, Inc., 2024. doi: 10.52202/079017-1787. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/661de50100d3115cf4317bb8b5219e56-Paper-Conference.pdf.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712, June 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 87400–87442, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d9e42b4d7163931f3689d6d6fbaa11d0-Paper-Conference.pdf.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 32111–32136. Curran Associates, Inc., 2024. doi: 10.52202/079017-1010. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/38b787fc530d0b31825827e2cc306656-Paper-Conference.pdf.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. 2025. URL https://arxiv.org/abs/2510.11690.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. 2024. URL https://arxiv.org/abs/2306.09305.

Appendix

Contents

A Additional Preliminaries and Related Work 15
A.1 Additional Preliminaries 15
A.2 More Discussion on Related Work 16

B Theoretical Analysis and Proofs 17
B.1 Proofs of Main Theoretical Results 17
B.2 Variance Analysis of JsFlow Design 19
B.3 Further Analysis 21

C Implementation Details 25
C.1 Comparison with Related Methods 25
C.2 JsFlow Design and Implementation 26
C.3 Training and Sampling Configurations 28

D Additional Experiments and Ablations 30
D.1 Extended Quantitative Results 30
D.2 Semantic Representation and Transport Schedule Ablations 30
D.3 Sampling Steps and Guidance Ablations 33

E More Visualization Results 34

A Additional Preliminaries and Related Work

The main paper presents the core Flow Matching formulation and the proposed augmented image–semantic transport. This section provides additional background of JsFlow.

A.1 Additional Preliminaries

Diffusion Transformer Backbones. Diffusion Transformers (DiT) parameterize generative models with a Vision Transformer operating on patchified image latents. Given an intermediate latent $\mathbf{x}_t \in \mathbb{R}^{L\times C_x}$ with L image tokens of dimension C_x, the model first projects the input into the Transformer hidden space:

$$\mathbf{h}_t = \mathbf{x}_t\mathbf{W}_{\text{emb}}, \qquad \mathbf{W}_{\text{emb}} \in \mathbb{R}^{C_x\times D}. \tag{17}$$

The Transformer processes $\mathbf{h}_t$ under timestep and class conditioning to produce $\mathbf{o}_t \in \mathbb{R}^{L\times D}$. For flow-based models, the final projection predicts the velocity field:

$$\mathbf{v}_\theta(\mathbf{x}_t, t, y) = \mathbf{o}_t\mathbf{W}_{\text{dec}}, \qquad \mathbf{W}_{\text{dec}} \in \mathbb{R}^{D\times C_x}. \tag{18}$$

SiT and LightningDiT emphasize two complementary aspects of generative Transformer design. SiT extends the DiT backbone to the stochastic-interpolant framework, learning continuous-time velocity fields and supporting both ODE- and SDE-based sampling. LightningDiT instead focuses on training efficiency by combining vision-aligned VAE latents with optimized training strategies and architectural designs.

Despite their different objectives, both models operate on sequences of latent image tokens and expose intermediate token representations throughout the Transformer blocks. This shared interface allows JsFlow to prepend semantic tokens and jointly predict image and semantic velocities without redesigning the backbone. It also enables direct representation-level supervision on intermediate hidden states.

Representation Alignment. REPA introduces an auxiliary objective that aligns the intermediate representations of a generative Transformer with visual features extracted from a frozen pretrained encoder. Given a clean image $\mathbf{x}$, the encoder produces patch-level target representations

$$\mathbf{R} = E_{\text{vis}}(\mathbf{x}) \in \mathbb{R}^{N\times D_R}, \tag{19}$$

where E_{vis} remains frozen during training, N denotes the number of visual tokens, and D_R is their feature dimension. For an intermediate image latent $\mathbf{X}_t$, let $\mathbf{H}_t^{(\ell)}$ denote the image-token representations produced by the ℓ-th Transformer block. A trainable projection head g_ϕ maps these representations into the feature space of the visual encoder:

$$\widehat{\mathbf{R}}_t^{(\ell)} = g_\phi\left(\mathbf{H}_t^{(\ell)}\right) \in \mathbb{R}^{N\times D_R}. \tag{20}$$

REPA encourages the projected hidden representations to match the clean encoder targets through a token-wise similarity objective:

$$\mathcal{L}_{\text{REPA}} = -\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\text{sim}\left(\mathbf{R}_n, \widehat{\mathbf{R}}_{t,n}^{(\ell)}\right)\right], \tag{21}$$

where $\text{sim}(\cdot,\cdot)$ is typically instantiated as cosine similarity. The alignment objective is jointly optimized with the original Flow Matching objective:

$$\mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}. \tag{22}$$

In JsFlow, REPA and joint semantic transport operate at different but complementary levels. REPA provides auxiliary representation-level supervision for intermediate image tokens, without introducing the target representations into the flow state or defining a transport path for them. In contrast, JsFlow explicitly incorporates semantic variables $\mathbf{C}_t$ into the joint generative state, transports them along dedicated semantic paths, and jointly predicts the image and semantic velocity fields. Thus, REPA improves the internal representations of the backbone, while joint semantic transport directly augments the generative state and reduces velocity ambiguity.

A.2 More Discussion on Related Work

Generative Models for Image Generation. Diffusion models learn to reverse a gradual noising process and have achieved strong image synthesis performance (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021). Score-based formulations connect this process to stochastic differential equations and an associated probability flow ODE (Song et al., 2021). Latent diffusion reduces computational cost by operating in a compressed image space (Rombach et al., 2022), while Transformer backbones improve scalability (Peebles & Xie, 2023; Ma et al., 2024). Flow Matching trains continuous normalizing flows through velocity regression along prescribed probability paths (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023). Rectified Flow uses straight-line interpolations between paired source and target samples; its reflow procedure progressively straightens learned trajectories to support sampling with fewer numerical integration steps (Liu et al., 2022). These ideas also underpin large-scale text-to-image generation (Esser et al., 2024).

Efficient Training of Diffusion and Flow Models. Recent work accelerates training through improved training strategies, latent representations, and supervision. FasterDiT improves training strategies and timestep supervision without changing the architecture (Yao et al., 2024), while LightningDiT combines vision-aligned latents with an efficient training recipe (Yao et al., 2025). Masked training reduces computation or strengthens representation learning (Zheng et al., 2024; Gao et al., 2023), and REPA and REPA-E exploit pretrained visual features through representation alignment and end-to-end VAE tuning (Yu et al., 2025; Leng et al., 2025). Contrastive Flow Matching introduces a contrastive objective to improve separation between conditional flows (Stoica et al., 2025). Joint image–feature modeling provides another route: REG and ReDi jointly generate image latents and semantic features (Wu et al., 2025; Kouzelis et al., 2025), while SFD coordinates semantic and texture generation through asynchronous schedules (Pan et al., 2025).

Semantic Guidance beyond Generation. Semantic information also supports visual tracking and test-time adaptation. SATATrack uses target language descriptions to guide temporal context propagation and applies online feature-distribution alignment for UAV tracking (Qiao et al., 2026b). For vision-language models, TPT adapts prompts through entropy minimization over confidence-selected augmented views (Shu et al., 2022), while TDA uses a dynamic key–value cache for adaptation without backpropagation (Karmanov et al., 2024). DPE jointly evolves visual and textual prototypes to accumulate task-specific knowledge (Zhang et al., 2024). BPRE further couples feature-quality assessment with prototype refinement through bidirectional feedback (Qiao et al., 2026a), and CPL-NC combines a class-aware prototype cache with negative contrastive learning to mitigate prototype degradation and confusion between similar classes (Qiao et al., 2025). Together, these methods explore complementary uses of semantic guidance, prompt adaptation, and prototype refinement to improve visual recognition under changing conditions.

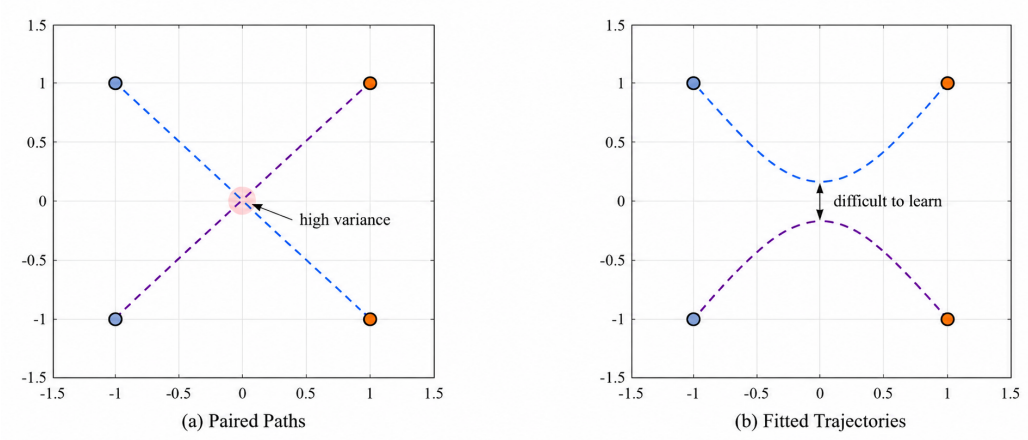


Figure 7: **Geometric interpretation of velocity variance.** (a) Intersecting paired paths induce conflicting velocity targets at the same intermediate state, leading to high conditional variance. (b) Consequently, fitting the velocity field around such ambiguous regions becomes more difficult.

B Theoretical Analysis and Proofs

B.1 Proofs of Main Theoretical Results

Theorem 1 (Bias–variance decomposition of Flow Matching). *Assume finite second moments and that the training randomness $\mathcal{D}$ is independent of the evaluation source–target pair. Let $\widehat{\theta}(\mathcal{D})$ denote the parameters obtained under $\mathcal{D}$, and write*

$$\widehat{\mathbf{v}}^X_{\mathcal{D}}(\mathbf{Z}_t, t) := \mathbf{v}^X_{\widehat{\theta}(\mathcal{D})}(\mathbf{Z}_t, t).$$

For squared image-velocity regression, the conditional prediction risk decomposes as

$$\mathbb{E}_{\mathcal{D},\,\mathbf{U}_X|\mathbf{Z}_t,t}\left[\left\|\widehat{\mathbf{v}}^X_{\mathcal{D}}(\mathbf{Z}_t,t) - \mathbf{U}_X(t)\right\|_2^2\right]$$
$$= \underbrace{\left\|\overline{\mathbf{v}}^X(\mathbf{Z}_t,t) - \mathbf{m}_Z(\mathbf{Z}_t,t)\right\|_2^2}_{\text{squared bias}} + \underbrace{\operatorname{Tr}\operatorname{Cov}_{\mathcal{D}}\left(\widehat{\mathbf{v}}^X_{\mathcal{D}}(\mathbf{Z}_t,t) \mid \mathbf{Z}_t,t\right)}_{\text{estimation variance}} + \underbrace{\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{Z}_t,t\right)}_{\text{irreducible velocity variance}},$$

where

$$\mathbf{m}_Z(\mathbf{Z}_t,t) := \mathbb{E}\left[\mathbf{U}_X(t) \mid \mathbf{Z}_t,t\right], \qquad \overline{\mathbf{v}}^X(\mathbf{Z}_t,t) := \mathbb{E}_{\mathcal{D}}\left[\widehat{\mathbf{v}}^X_{\mathcal{D}}(\mathbf{Z}_t,t)\right].$$

Proof. Fix $(\mathbf{Z}_t, t)$ and abbreviate

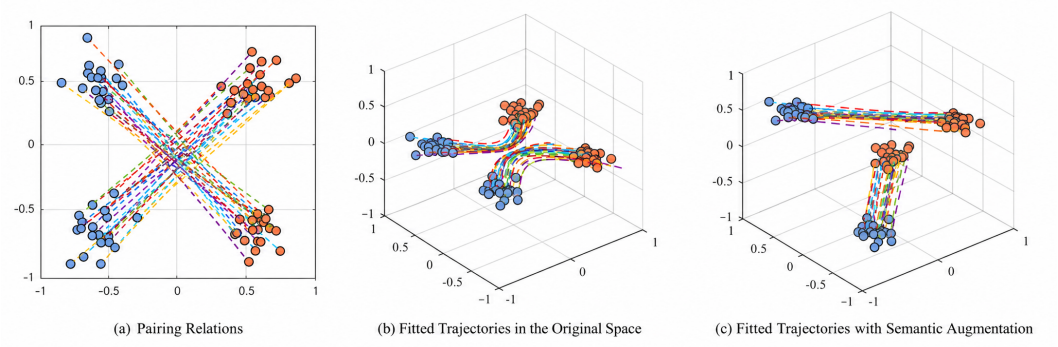
$$\widehat{\mathbf{V}}_{\mathcal{D}} := \widehat{\mathbf{v}}^X_{\mathcal{D}}(\mathbf{Z}_t,t), \qquad \mathbf{U} := \mathbf{U}_X(t),$$

and write

$$\overline{\mathbf{V}} := \mathbb{E}_{\mathcal{D}}[\widehat{\mathbf{V}}_{\mathcal{D}}], \qquad \mathbf{m} := \mathbb{E}[\mathbf{U}].$$

All expectations and covariances below are conditional on $(\mathbf{Z}_t, t)$. Then

$$\begin{aligned}
&\mathbb{E}\left[\|\widehat{\mathbf{V}}_{\mathcal{D}} - \mathbf{U}\|_2^2\right] \\
&= \mathbb{E}\left[\|(\widehat{\mathbf{V}}_{\mathcal{D}} - \mathbf{m}) + (\mathbf{m} - \mathbf{U})\|_2^2\right] \\
&= \mathbb{E}\left[\|\widehat{\mathbf{V}}_{\mathcal{D}} - \mathbf{m}\|_2^2\right] + \mathbb{E}\left[\|\mathbf{U} - \mathbf{m}\|_2^2\right] && // \ \mathbb{E}[\mathbf{U} - \mathbf{m} \mid \mathcal{D}] = \mathbf{0} \\
&= \|\overline{\mathbf{V}} - \mathbf{m}\|_2^2 + \mathbb{E}\left[\|\widehat{\mathbf{V}}_{\mathcal{D}} - \overline{\mathbf{V}}\|_2^2\right] + \mathbb{E}\left[\|\mathbf{U} - \mathbf{m}\|_2^2\right] && // \ \mathbb{E}_{\mathcal{D}}[\widehat{\mathbf{V}}_{\mathcal{D}} - \overline{\mathbf{V}}] = \mathbf{0} \\
&= \|\overline{\mathbf{V}} - \mathbf{m}\|_2^2 + \operatorname{Tr}\operatorname{Cov}_{\mathcal{D}}(\widehat{\mathbf{V}}_{\mathcal{D}}) + \operatorname{Tr}\operatorname{Cov}(\mathbf{U}). && // \ \mathbb{E}\|\mathbf{Y} - \mathbb{E}\mathbf{Y}\|_2^2 = \operatorname{Tr}\operatorname{Cov}(\mathbf{Y})
\end{aligned} \tag{23}$$



(a) Pairing Relations (b) Fitted Trajectories in the Original Space (c) Fitted Trajectories with Semantic Augmentation

Figure 8: **Geometric interpretation of semantic disambiguation through state augmentation.** (a) In the original image space, different source–target pairings can intersect at the same intermediate state, producing ambiguous velocity targets. (b) After augmenting the state with an additional semantic dimension, the paired trajectories are lifted into a higher-dimensional space and become separable. (c) Consequently, the fitted trajectories in the augmented space no longer suffer from crossing-induced ambiguity, leading to a simpler and more learnable velocity field

The first two terms are nonnegative and vanish when $\widehat{\mathbf{V}}_{\mathcal{D}} = \mathbf{m}$ almost surely. Hence, among all square-integrable predictors of $(\mathbf{Z}_t, t)$, the conditional risk is minimized by

$$\mathbf{v}_X^\star(\mathbf{Z}_t, t) = \mathbf{m}_Z(\mathbf{Z}_t, t) = \mathbb{E}\left[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right],$$

with minimum

$$\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right).$$

Averaging over $\mathbf{Z}_t$ at fixed t gives Eq. (9). □

Theorem 2 (Semantic augmentation reduces velocity ambiguity). *Let* $\mathbf{Z}_t = [\mathbf{X}_t; \mathbf{C}_t]$, *and define*

$$\mathcal{V}_X(t) := \mathbb{E}\left[\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{X}_t, t\right)\right], \qquad \mathcal{V}_Z(t) := \mathbb{E}\left[\operatorname{Tr}\operatorname{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right)\right].$$

Then the reduction in the minimum achievable image-velocity regression risk is

$$\Delta\mathcal{L}^\star_{\text{image}}(t) := \mathcal{V}_X(t) - \mathcal{V}_Z(t) = \mathbb{E}\left[\left\|\mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t] - \mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{X}_t, t]\right\|_2^2\right] \geq 0. \tag{24}$$

Proof. Fix t and define

$$\mathbf{m}_X := \mathbb{E}[\mathbf{U} \mid \mathbf{X}_t, t], \qquad \mathbf{m}_Z := \mathbb{E}[\mathbf{U} \mid \mathbf{Z}_t, t].$$

For any square-integrable vector $\mathbf{Y}$, denote

$$\mathrm{D}(\mathbf{Y}) := \operatorname{Tr}\operatorname{Cov}(\mathbf{Y}), \qquad \mathrm{D}(\mathbf{Y} \mid \mathbf{A}) := \operatorname{Tr}\operatorname{Cov}(\mathbf{Y} \mid \mathbf{A}).$$

Since $\mathbf{Z}_t$ contains $\mathbf{X}_t$, we have $\mathbb{E}[\mathbf{m}_Z \mid \mathbf{X}_t, t] = \mathbf{m}_X$. Therefore,

$$\begin{aligned}
&\mathcal{V}_X(t) - \mathcal{V}_Z(t) \\
&= \mathbb{E}[\mathrm{D}(\mathbf{U} \mid \mathbf{X}_t, t)] - \mathbb{E}[\mathrm{D}(\mathbf{U} \mid \mathbf{Z}_t, t)] \\
&= \mathrm{D}(\mathbf{m}_Z) - \mathrm{D}(\mathbf{m}_X) && // \ \mathrm{D}(\mathbf{Y}) = \mathbb{E}[\mathrm{D}(\mathbf{Y} \mid \mathbf{A})] + \mathrm{D}(\mathbb{E}[\mathbf{Y} \mid \mathbf{A}]) \\
&= \mathbb{E}[\mathrm{D}(\mathbf{m}_Z \mid \mathbf{X}_t, t)] && // \ \mathbb{E}[\mathbf{m}_Z \mid \mathbf{X}_t, t] = \mathbf{m}_X \\
&= \mathbb{E}\left[\|\mathbf{m}_Z - \mathbf{m}_X\|_2^2\right] && // \ \mathbb{E}\|\mathbf{Y} - \mathbb{E}[\mathbf{Y} \mid \mathbf{A}]\|_2^2 = \mathbb{E}[\mathrm{D}(\mathbf{Y} \mid \mathbf{A})] \\
&= \mathbb{E}\left[\left\|\mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t] - \mathbb{E}[\mathbf{U}_X(t) \mid \mathbf{X}_t, t]\right\|_2^2\right] \geq 0. && (25)
\end{aligned}$$

Thus, $\mathcal{V}_Z(t) \leq \mathcal{V}_X(t)$, with equality if and only if $\mathbf{m}_Z = \mathbf{m}_X$ almost surely. Equivalently, the reduction is strict precisely when $\mathbb{P}(\mathbf{m}_Z \neq \mathbf{m}_X) > 0$. □

Theorem 2 further yields two special cases.

Deterministic semantic coordinates. If $\mathbf{C}_t = g_t(\mathbf{X}_t)$ almost surely for a measurable function g_t, then

$$\mathbf{m}_Z = \mathbb{E}[\mathbf{U} \mid \mathbf{X}_t, g_t(\mathbf{X}_t), t] = \mathbb{E}[\mathbf{U} \mid \mathbf{X}_t, t] = \mathbf{m}_X \qquad \text{a.s.}$$

Hence,

$$\mathcal{V}_Z(t) = \mathcal{V}_X(t).$$

Thus, semantics that are fully determined by the current image state cannot reduce the irreducible velocity ambiguity.

Complete velocity disambiguation. At the opposite extreme,

$$\begin{aligned}
\mathcal{V}_Z(t) = 0 &\iff \mathbb{E}\left[\|\mathbf{U} - \mathbf{m}_Z\|_2^2\right] = 0 && // \ \mathcal{V}_Z(t) = \mathbb{E}\|\mathbf{U} - \mathbf{m}_Z\|_2^2 \\
&\iff \mathbf{U} = \mathbf{m}_Z \qquad \text{a.s.} \\
&\iff \exists\, g_t \text{ measurable} : \mathbf{U} = g_t(\mathbf{Z}_t) \qquad \text{a.s.} \quad // \text{ Doob–Dynkin lemma}
\end{aligned} \tag{26}$$

Hence, $\mathcal{V}_Z(t) = 0$ exactly when the augmented state fully determines the image velocity.

The same augmented-matrix formulation also covers standard class-conditional generation. Let $\mathbf{Y}$ denote the representation of a class label, and define the augmented endpoints as

$$\widetilde{\mathbf{Z}}_0 = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{Y} \\ 0 \end{bmatrix}, \qquad \widetilde{\mathbf{Z}}_1 = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y} \\ 1 \end{bmatrix}.$$

Using the block-diagonal interpolation operator

$$\widetilde{\mathbf{\Lambda}}(t) = \begin{bmatrix} \alpha(t)\mathbf{I}_X & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma(t)\mathbf{I}_Y & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} & t \end{bmatrix},$$

the augmented path is written as

$$\widetilde{\mathbf{Z}}_t = \big(\mathbf{I} - \widetilde{\mathbf{\Lambda}}(t)\big)\widetilde{\mathbf{Z}}_0 + \widetilde{\mathbf{\Lambda}}(t)\widetilde{\mathbf{Z}}_1 = \begin{bmatrix} (1-\alpha(t))\mathbf{X}_0 + \alpha(t)\mathbf{X}_1 \\ (1-\gamma(t))\mathbf{Y} + \gamma(t)\mathbf{Y} \\ 1 - t + t \end{bmatrix} = \begin{bmatrix} \mathbf{X}_t \\ \mathbf{Y} \\ t \end{bmatrix}.$$

Differentiating this path gives the corresponding augmented ordinary differential equation,

$$\frac{\mathrm{d}\widetilde{\mathbf{Z}}_t}{\mathrm{d}t} = \dot{\widetilde{\mathbf{\Lambda}}}(t)\big(\widetilde{\mathbf{Z}}_1 - \widetilde{\mathbf{Z}}_0\big) = \begin{bmatrix} \dot{\alpha}(t)(\mathbf{X}_1 - \mathbf{X}_0) \\ \dot{\gamma}(t)(\mathbf{Y} - \mathbf{Y}) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{U}_X(t) \\ \mathbf{0} \\ 1 \end{bmatrix}.$$

Thus, the class label enters as a fixed coordinate of the augmented state, while the time coordinate evolves with unit velocity. Standard class-conditional generation can therefore be described within the same augmented-matrix framework. Therefore, the performance gains from conditional generation can also be explained by the reduction in conditional variance.

B.2 VARIANCE ANALYSIS OF JSFLOW DESIGN

In this section, we analyze how the design choices in JsFlow affect velocity variance.

B.2.1 MULTI-VIEW SEMANTIC AUGMENTATION.

We first analyze multi-view semantic augmentation. The following corollary shows that successively incorporating complementary semantic representations monotonically reduces the conditional variance of the target image velocity.

Corollary 1 (Monotonic reduction with complementary semantics). *Let*

$$\mathbf{Z}_t^{(k)} = \left[\mathbf{X}_t^\top, \mathbf{C}_t^{(1)\top}, \ldots, \mathbf{C}_t^{(k)\top}\right]^\top. \tag{27}$$

Then

$$\mathcal{V}_X(t) \geq \mathcal{V}_{Z^{(1)}}(t) \geq \mathcal{V}_{Z^{(2)}}(t) \geq \cdots \geq \mathcal{V}_{Z^{(K)}}(t) \geq 0. \tag{28}$$

Proof. Let

$$\mathbf{Z}_t^{(0)} := \mathbf{X}_t, \qquad \mathbf{m}_k := \mathbb{E}\left[\mathbf{U} \mid \mathbf{Z}_t^{(k)}, t\right], \qquad k = 0, \ldots, K.$$

Since $\mathbf{Z}_t^{(k)}$ is obtained by augmenting $\mathbf{Z}_t^{(k-1)}$ with $\mathbf{C}_t^{(k)}$, Theorem 2 gives

$$\mathcal{V}_{Z^{(k-1)}}(t) - \mathcal{V}_{Z^{(k)}}(t) = \mathbb{E}\left[\|\mathbf{m}_k - \mathbf{m}_{k-1}\|_2^2\right] \geq 0, \qquad k = 1, \ldots, K. \tag{29}$$

With $\mathcal{V}_{Z^{(0)}}(t) = \mathcal{V}_X(t)$, repeated application yields

$$\mathcal{V}_X(t) \geq \mathcal{V}_{Z^{(1)}}(t) \geq \mathcal{V}_{Z^{(2)}}(t) \geq \cdots \geq \mathcal{V}_{Z^{(K)}}(t) \geq 0. \tag{30}$$

Moreover, the k-th reduction is strict if and only if

$$\mathbb{P}\left(\mathbf{m}_k \neq \mathbf{m}_{k-1}\right) > 0.$$

Thus, an additional semantic coordinate strictly reduces velocity ambiguity exactly when it provides velocity-relevant information beyond the preceding augmented state. □

B.2.2 SEMANTIC-FIRST ASYNCHRONOUS TRANSPORT.

We next examine how the semantic transport schedule affects velocity ambiguity. The following proposition shows that revealing the semantic endpoint earlier reduces uncertainty in the image velocity.

Proposition 1 (Semantic-first transport reduces velocity ambiguity). *Consider*

$$\mathbf{X}_t = (1 - \alpha(t))\mathbf{X}_0 + \alpha(t)\mathbf{X}_1, \qquad \mathbf{C}_t^{(j)} = (1 - \beta_j(t))\mathbf{C}_0^{(j)} + \beta_j(t)\mathbf{C}_1, \quad j \in \{1, 2\}, \tag{31}$$

where the Gaussian sources are mutually independent and independent of $(\mathbf{X}_1, \mathbf{C}_1)$. *If* $0 \leq \beta_1(t) \leq \beta_2(t) \leq 1$, *then*

$$\mathcal{V}_{\beta_2}(t) \leq \mathcal{V}_{\beta_1}(t), \tag{32}$$

where

$$\mathcal{V}_{\beta_j}(t) := \mathbb{E}\left[\mathrm{Tr\,Cov}\left(\mathbf{U}_X(t) \mid \mathbf{X}_t, \mathbf{C}_t^{(j)}, t\right)\right]. \tag{33}$$

Proof. For $\alpha(t) < 1$, the image velocity can be written as

$$\mathbf{U}_X(t) = \frac{\dot{\alpha}(t)}{1 - \alpha(t)}\big(\mathbf{X}_1 - \mathbf{X}_t\big). \tag{34}$$

Therefore,

$$\mathcal{V}_{\beta_j}(t) = \left(\frac{\dot{\alpha}(t)}{1 - \alpha(t)}\right)^2 \mathbb{E}\left[\mathrm{Tr\,Cov}\left(\mathbf{X}_1 \mid \mathbf{X}_t, \mathbf{C}_t^{(j)}, t\right)\right]. \tag{35}$$

Thus, image-velocity ambiguity is proportional to the remaining uncertainty of the image endpoint.

Fix t and write $b_j = \beta_j(t)$. For $b_2 > 0$, define

$$\widetilde{\mathbf{C}}_t^{(1)} = \frac{b_1}{b_2}\mathbf{C}_t^{(2)} + \left[(1 - b_1)^2 - \frac{b_1^2}{b_2^2}(1 - b_2)^2\right]^{1/2} \boldsymbol{\epsilon}, \tag{36}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent of all other variables. Since $b_1 \leq b_2$, the variance of the added noise is nonnegative. By construction, $\widetilde{\mathbf{C}}_t^{(1)}$ has the same distribution as $\mathbf{C}_t^{(1)}$. Hence, the semantic state under β_1 is statistically equivalent to a noisy version of the semantic state under β_2.

The conditional variance decomposition therefore gives

$$\mathbb{E}\left[\mathrm{Tr\,Cov}\left(\mathbf{X}_1 \mid \mathbf{X}_t, \mathbf{C}_t^{(2)}, t\right)\right] \leq \mathbb{E}\left[\mathrm{Tr\,Cov}\left(\mathbf{X}_1 \mid \mathbf{X}_t, \mathbf{C}_t^{(1)}, t\right)\right]. \tag{37}$$

Combining this inequality with Eq. (35) proves the result. The inequality is strict when the additional semantic information revealed by $\beta_2(t)$ improves the prediction of $\mathbf{X}_1$. □

B.3 Further Analysis

B.3.1 From Velocity Ambiguity to Gradient Variance

The preceding results characterize the irreducible ambiguity of the image velocity. We next show how this ambiguity propagates to stochastic gradients during training.

Proposition 2 (Velocity ambiguity induces gradient variance). *Consider the per-sample image-velocity objective*

$$\ell_\theta = \frac{1}{2} \left\| \mathbf{v}_\theta^X(\mathbf{Z}_t, t) - \mathbf{U}_X(t) \right\|_2^2, \tag{38}$$

and let

$$\mathbf{J}_\theta(\mathbf{Z}_t, t) := \frac{\partial \mathbf{v}_\theta^X(\mathbf{Z}_t, t)}{\partial \theta} \in \mathbb{R}^{d_x \times p}. \tag{39}$$

Then the target-induced conditional gradient covariance is

$$\mathrm{Cov}\left(\nabla_\theta \ell_\theta \mid \mathbf{Z}_t, t\right) = \mathbf{J}_\theta^\top \mathbf{\Sigma}_Z \mathbf{J}_\theta, \qquad \mathbf{\Sigma}_Z := \mathrm{Cov}\left(\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right). \tag{40}$$

Consequently,

$$\mathrm{Tr}\,\mathrm{Cov}\left(\nabla_\theta \ell_\theta \mid \mathbf{Z}_t, t\right) \le \|\mathbf{J}_\theta\|_{\mathrm{op}}^2 \,\mathrm{Tr}\,\mathbf{\Sigma}_Z. \tag{41}$$

Proof. Conditioned on $(\mathbf{Z}_t, t)$, both $\mathbf{v}_\theta^X(\mathbf{Z}_t, t)$ and $\mathbf{J}_\theta(\mathbf{Z}_t, t)$ are deterministic. Writing

$$\mathbf{m}_Z := \mathbb{E}\left[\mathbf{U}_X(t) \mid \mathbf{Z}_t, t\right],$$

the centered per-sample gradient is

$$\nabla_\theta \ell_\theta - \mathbb{E}\left[\nabla_\theta \ell_\theta \mid \mathbf{Z}_t, t\right] = -\mathbf{J}_\theta^\top \left(\mathbf{U}_X(t) - \mathbf{m}_Z\right). \tag{42}$$

Taking the conditional covariance gives Eq. (40). Moreover,

$$\mathrm{Tr}\left(\mathbf{J}_\theta^\top \mathbf{\Sigma}_Z \mathbf{J}_\theta\right) = \mathrm{Tr}\left(\mathbf{\Sigma}_Z \mathbf{J}_\theta \mathbf{J}_\theta^\top\right) \le \|\mathbf{J}_\theta\|_{\mathrm{op}}^2 \,\mathrm{Tr}\,\mathbf{\Sigma}_Z, \tag{43}$$

which proves Eq. (41). □

If $\|\mathbf{J}_\theta(\mathbf{Z}_t, t)\|_{\mathrm{op}} \le L$ almost surely, averaging Eq. (41) over $\mathbf{Z}_t$ gives

$$\mathbb{E}\left[\mathrm{Tr}\,\mathrm{Cov}\left(\nabla_\theta \ell_\theta \mid \mathbf{Z}_t, t\right)\right] \le L^2 \mathcal{V}_Z(t). \tag{44}$$

Thus, the reduction in $\mathcal{V}_Z(t)$ established by Theorem 2 also reduces an upper bound on the target-induced gradient variance, provided that the Jacobian amplification remains comparable.

This connection can also be expressed in terms of gradient disagreement. Let $\mathbf{U}_X^{(1)}(t)$ and $\mathbf{U}_X^{(2)}(t)$ be conditionally independent samples from $p(\mathbf{U}_X(t) \mid \mathbf{Z}_t, t)$, and let $\mathbf{g}_\theta^{(1)}$ and $\mathbf{g}_\theta^{(2)}$ be their corresponding gradients at the same state. Then

$$\mathbb{E}\left[\left\|\mathbf{g}_\theta^{(1)} - \mathbf{g}_\theta^{(2)}\right\|_2^2 \mid \mathbf{Z}_t, t\right] = 2\,\mathrm{Tr}\left(\mathbf{J}_\theta^\top \mathbf{\Sigma}_Z \mathbf{J}_\theta\right). \tag{45}$$

Hence, velocity targets compatible with the same observed state can produce conflicting gradients, while semantic augmentation reduces the target-side ambiguity responsible for this disagreement.

These results characterize the target-induced component of gradient variance. They do not imply an unconditional ordering of the complete SGD covariance between different architectures, since state augmentation may also change the network Jacobian, optimization trajectory, and minibatch statistics.

The preceding covariance identity is local to each observed state. We now average over the training distribution and show that velocity ambiguity enters explicitly into a standard convergence bound for stochastic gradient descent.

Let

$$\mathcal{L}(\theta) := \mathbb{E}\left[\ell_\theta\right], \qquad \overline{\mathcal{V}}_Z := \mathbb{E}_t\left[\mathcal{V}_Z(t)\right], \tag{46}$$

and define the conditional mean gradient

$$\overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t) := \mathbb{E}\left[\nabla_\theta \ell_\theta \mid \mathbf{Z}_t, t\right]. \tag{47}$$

Corollary 2(Velocity variance controls the stochastic convergence term). *Assume that $\mathcal{L}$ is $L_\mathcal{L}$-smooth and bounded below by $\mathcal{L}^\star$. Suppose that, along the optimization trajectory,*

$$\left\|\mathbf{J}_\theta(\mathbf{Z}_t, t)\right\|_{\mathrm{op}} \le G, \tag{48}$$

and that the gradient variation caused by sampling different observed states is bounded as

$$\mathbb{E}\left[\left\|\overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t) - \nabla\mathcal{L}(\theta)\right\|_2^2\right] \le \sigma_{\mathrm{state}}^2. \tag{49}$$

Consider minibatch SGD with batch size B,

$$\theta_{k+1} = \theta_k - \eta \mathbf{g}_k, \qquad \mathbb{E}\left[\mathbf{g}_k \mid \theta_k\right] = \nabla\mathcal{L}(\theta_k). \tag{50}$$

Then, for any constant step size $\eta \le 1/L_\mathcal{L}$,

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_k)\right\|_2^2\right] \le \frac{2\big(\mathcal{L}(\theta_0) - \mathcal{L}^\star\big)}{\eta T} + \frac{L_\mathcal{L}\eta}{B}\left(\sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z\right). \tag{51}$$

Proof. For a single-sample gradient $\mathbf{g}_\theta := \nabla_\theta \ell_\theta$, write

$$\mathbf{g}_\theta - \nabla\mathcal{L}(\theta) = \left(\mathbf{g}_\theta - \overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t)\right) + \left(\overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t) - \nabla\mathcal{L}(\theta)\right). \tag{52}$$

The first term has zero conditional mean given $(\mathbf{Z}_t, t)$, so the cross term vanishes after taking expectation. Therefore,

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{g}_\theta - \nabla\mathcal{L}(\theta)\right\|_2^2\right] &= \mathbb{E}\left[\left\|\mathbf{g}_\theta - \overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t)\right\|_2^2\right] + \mathbb{E}\left[\left\|\overline{\mathbf{g}}_\theta(\mathbf{Z}_t, t) - \nabla\mathcal{L}(\theta)\right\|_2^2\right] \\ &\le G^2\overline{\mathcal{V}}_Z + \sigma_{\mathrm{state}}^2,\end{aligned} \tag{53}$$

where the inequality follows from Eqs. (44) and (49).

For a minibatch of B independent samples, Eq. (53) gives

$$\mathbb{E}\left[\left\|\mathbf{g}_k - \nabla\mathcal{L}(\theta_k)\right\|_2^2 \mid \theta_k\right] \le \frac{\sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z}{B}. \tag{54}$$

By the $L_\mathcal{L}$-smoothness of $\mathcal{L}$,

$$\mathcal{L}(\theta_{k+1}) \le \mathcal{L}(\theta_k) - \eta\left\langle\nabla\mathcal{L}(\theta_k), \mathbf{g}_k\right\rangle + \frac{L_\mathcal{L}\eta^2}{2}\left\|\mathbf{g}_k\right\|_2^2. \tag{55}$$

Taking conditional expectation and using the unbiasedness of $\mathbf{g}_k$ together with Eq. (54) yields

$$\mathbb{E}\left[\mathcal{L}(\theta_{k+1}) \mid \theta_k\right] \le \mathcal{L}(\theta_k) - \eta\left(1 - \frac{L_\mathcal{L}\eta}{2}\right)\left\|\nabla\mathcal{L}(\theta_k)\right\|_2^2 + \frac{L_\mathcal{L}\eta^2}{2B}\left(\sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z\right). \tag{56}$$

Since $\eta \le 1/L_\mathcal{L}$,

$$\frac{\eta}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_k)\right\|_2^2\right] \le \mathbb{E}\left[\mathcal{L}(\theta_k) - \mathcal{L}(\theta_{k+1})\right] + \frac{L_\mathcal{L}\eta^2}{2B}\left(\sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z\right). \tag{57}$$

Summing over $k = 0, \ldots, T-1$ and using $\mathcal{L}(\theta_T) \ge \mathcal{L}^\star$ proves Eq. (51). □

Let

$$\Delta_0 := \mathcal{L}(\theta_0) - \mathcal{L}^\star, \qquad \sigma_Z^2 := \sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z. \tag{58}$$

Choosing

$$\eta = \min\left\{\frac{1}{L_\mathcal{L}}, \sqrt{\frac{2B\Delta_0}{L_\mathcal{L}\sigma_Z^2 T}}\right\} \tag{59}$$

gives

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_k)\right\|_2^2\right] = \mathcal{O}\left(\frac{L_\mathcal{L}\Delta_0}{T} + \sqrt{\frac{L_\mathcal{L}\Delta_0\left(\sigma_{\mathrm{state}}^2 + G^2\overline{\mathcal{V}}_Z\right)}{BT}}\right). \tag{60}$$

Hence, under comparable smoothness, Jacobian amplification, and state-sampling variance, reducing $\overline{\mathcal{V}}_Z$ improves the stochastic term in the SGD convergence bound.

B.3.2 Geometric Characterization of Semantic Disambiguation

The probabilistic conditions above also admit a local geometric interpretation. Let

$$\mathbf{A} := \big(\mathbf{Z}_0, \mathbf{Z}_1, \boldsymbol{\xi}\big) \in \mathcal{A}, \qquad \mathbf{Z}_i = \begin{bmatrix} \mathbf{X}_i \\ \mathbf{C}_i \end{bmatrix}, \quad i \in \{0, 1\}, \tag{61}$$

collect the source and target endpoints together with any additional randomness $\boldsymbol{\xi}$ required to determine a conditional trajectory. When the trajectory is fully determined by its endpoints, $\boldsymbol{\xi}$ can be omitted.

For a realization $\mathbf{a} \in \mathcal{A}$ and a fixed time t, define the image-state, semantic-state, and image-velocity maps as

$$F_t(\mathbf{a}) := \mathbf{X}_t(\mathbf{a}), \qquad H_t(\mathbf{a}) := \mathbf{C}_t(\mathbf{a}), \qquad W_t(\mathbf{a}) := \mathbf{U}_X(t; \mathbf{a}). \tag{62}$$

Image-invisible velocity ambiguity. An infinitesimal perturbation $\delta\mathbf{a}$ is invisible from the current image state but relevant to its velocity if

$$DF_t(\mathbf{a})\delta\mathbf{a} = \mathbf{0}, \qquad DW_t(\mathbf{a})\delta\mathbf{a} \neq \mathbf{0}. \tag{63}$$

Such a direction represents locally indistinguishable image states that require different image-velocity targets.

Ambiguity-resolving semantic feature. The semantic coordinate resolves the ambiguous direction in Eq. (63) when

$$DH_t(\mathbf{a})\delta\mathbf{a} \neq \mathbf{0}. \tag{64}$$

Thus, an informative semantic feature must respond to velocity-relevant variations that remain invisible from $\mathbf{X}_t$ alone.

Locally velocity-sufficient semantic feature. All local velocity ambiguity is resolved when

$$\ker\big(DF_t(\mathbf{a})\big) \cap \ker\big(DH_t(\mathbf{a})\big) \subseteq \ker\big(DW_t(\mathbf{a})\big). \tag{65}$$

Under this condition, every perturbation that remains invisible after observing both $\mathbf{X}_t$ and $\mathbf{C}_t$ also leaves the image velocity unchanged. This is the local geometric counterpart of velocity sufficiency.

Image-derived semantic feature. Suppose that the semantic-state map factors through the image-state map:

$$H_t = h_t \circ F_t \tag{66}$$

for some differentiable function h_t. Equivalently, for every $\mathbf{a} \in \mathcal{A}$,

$$\mathbf{C}_t(\mathbf{a}) = h_t\big(\mathbf{X}_t(\mathbf{a})\big). \tag{67}$$

The chain rule gives

$$DH_t(\mathbf{a}) = Dh_t\big(F_t(\mathbf{a})\big)DF_t(\mathbf{a}). \tag{68}$$

Consequently,

$$DF_t(\mathbf{a})\delta\mathbf{a} = \mathbf{0} \quad \Longrightarrow \quad DH_t(\mathbf{a})\delta\mathbf{a} = \mathbf{0}. \tag{69}$$

Therefore, a semantic feature computed only from the current image state cannot separate source–target trajectories that produce exactly the same $\mathbf{X}_t$. This recovers the probabilistic result that deterministic features of $\mathbf{X}_t$ cannot reduce the irreducible velocity variance.

Complementary semantic views. For semantic views $H_t^{(1)}, \ldots, H_t^{(k)}$, define the directions that remain invisible to the image state and the first k semantic views as

$$\mathcal{K}_t^{(k)}(\mathbf{a}) := \ker\big(DF_t(\mathbf{a})\big) \cap \bigcap_{j=1}^{k} \ker\big(DH_t^{(j)}(\mathbf{a})\big). \tag{70}$$

A new view $H_t^{(k+1)}$ is locally complementary if there exists $\delta\mathbf{a} \in \mathcal{K}_t^{(k)}(\mathbf{a})$ such that

$$DW_t(\mathbf{a})\delta\mathbf{a} \neq \mathbf{0}, \qquad DH_t^{(k+1)}(\mathbf{a})\delta\mathbf{a} \neq \mathbf{0}. \tag{71}$$

In other words, the new view exposes a velocity-relevant direction that remains invisible to the image state and all preceding semantic views.

A collection of K semantic views completely resolves local velocity ambiguity when

$$\ker\big(DF_t(\mathbf{a})\big) \cap \bigcap_{k=1}^{K} \ker\big(DH_t^{(k)}(\mathbf{a})\big) \subseteq \ker\big(DW_t(\mathbf{a})\big). \tag{72}$$

This provides a local geometric interpretation of the probabilistic condition that a semantic feature reduces velocity variance only when it reveals additional velocity-relevant information.

B.3.3 Fisher-Information Characterization of Velocity Variance

Under the standard Gaussian source and linear interpolation, the irreducible velocity variance admits an exact characterization in terms of Fisher information. This characterization makes explicit how semantic conditioning reduces velocity ambiguity.

Fisher characterization. *Assume*

$$\mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \qquad \mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1, \qquad \mathbf{U}_X(t) = \mathbf{X}_1 - \mathbf{X}_0, \tag{73}$$

where $\mathbf{X}_0$ is independent of $(\mathbf{X}_1, \mathbf{C}_t)$. Assume further that the relevant marginal and conditional densities are differentiable and have finite Fisher information. Define

$$\mathcal{J}_X(t) := \mathbb{E}\left[\left\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\big|_{\mathbf{x}=\mathbf{X}_t}\right\|_2^2\right], \quad \mathcal{J}_{X|C}(t) := \mathbb{E}\left[\left\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{C}_t)\big|_{\mathbf{x}=\mathbf{X}_t}\right\|_2^2\right]. \tag{74}$$

Then, for $t \in (0,1)$,

$$\mathcal{V}_X(t) = \frac{1}{t^2}\left[d - (1-t)^2 \mathcal{J}_X(t)\right], \quad \mathcal{V}_Z(t) = \frac{1}{t^2}\left[d - (1-t)^2 \mathcal{J}_{X|C}(t)\right]. \tag{75}$$

Proof. The conditional score of the Gaussian perturbation satisfies

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{C}_t)\big|_{\mathbf{x}=\mathbf{X}_t} = -\frac{1}{1-t}\mathbb{E}\left[\mathbf{X}_0 \mid \mathbf{X}_t, \mathbf{C}_t, t\right]. \tag{76}$$

Moreover, the linear path gives

$$\mathbf{U}_X(t) = \frac{\mathbf{X}_t - \mathbf{X}_0}{t}. \tag{77}$$

Hence,

$$\begin{aligned}
\mathcal{V}_Z(t) &= \frac{1}{t^2}\mathbb{E}\left[\mathrm{Tr}\,\mathrm{Cov}\left(\mathbf{X}_0 \mid \mathbf{X}_t, \mathbf{C}_t, t\right)\right] \\
&= \frac{1}{t^2}\left\{\mathbb{E}\left[\|\mathbf{X}_0\|_2^2\right] - \mathbb{E}\left[\|\mathbb{E}\left[\mathbf{X}_0 \mid \mathbf{X}_t, \mathbf{C}_t, t\right]\|_2^2\right]\right\} \\
&= \frac{1}{t^2}\left[d - (1-t)^2 \mathcal{J}_{X|C}(t)\right],
\end{aligned} \tag{78}$$

where the last equality follows from Eq. (76) and $\mathbb{E}[\|\mathbf{X}_0\|_2^2] = d$. Removing $\mathbf{C}_t$ from the conditioning gives Eq. (75). □

Eq. (75) gives the exact variance reduction

$$\mathcal{V}_X(t) - \mathcal{V}_Z(t) = \frac{(1-t)^2}{t^2}\left[\mathcal{J}_{X|C}(t) - \mathcal{J}_X(t)\right]. \tag{79}$$

The Fisher-information difference also admits a direct semantic interpretation. By Bayes' rule,

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{c}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{c} \mid \mathbf{x}). \tag{80}$$

The cross term vanishes after taking expectation because

$$\mathbb{E}\left[\nabla_{\mathbf{x}} \log p_t(\mathbf{C}_t \mid \mathbf{x}) \mid \mathbf{X}_t = \mathbf{x}\right] = \mathbf{0}. \tag{81}$$

Consequently,

$$\mathcal{J}_{X|C}(t) - \mathcal{J}_X(t) = \mathbb{E}\left[\left\|\nabla_{\mathbf{x}} \log p_t(\mathbf{C}_t \mid \mathbf{x})\big|_{\mathbf{x}=\mathbf{X}_t}\right\|_2^2\right] \geq 0. \tag{82}$$

Combining Eqs. (79) and (82) yields

$$\mathcal{V}_X(t) - \mathcal{V}_Z(t) = \frac{(1-t)^2}{t^2}\mathbb{E}\left[\left\|\nabla_{\mathbf{x}} \log p_t(\mathbf{C}_t \mid \mathbf{x})\big|_{\mathbf{x}=\mathbf{X}_t}\right\|_2^2\right] \geq 0. \tag{83}$$

Equations (75)–(83) show that the velocity variance is determined by three immediate quantities: the image dimension d, the timestep-dependent scaling, and the Fisher information of the observed state. For the image-only state, the relevant quantity is $\mathcal{J}_X(t)$; after semantic augmentation, it becomes $\mathcal{J}_{X|C}(t)$.

Therefore, the velocity variance is jointly determined by the source distribution, the target data distribution, the source–target coupling, and the interpolation path. In JsFlow, it is further affected by the semantic representation and the semantic transport schedule, which determine how much additional information $\mathbf{C}_t$ provides about the ambiguous image velocity. This contribution is quantified by the conditional Fisher-information gain $\mathcal{J}_{X|C}(t) - \mathcal{J}_X(t)$.

C Implementation Details

C.1 Comparison with Related Methods

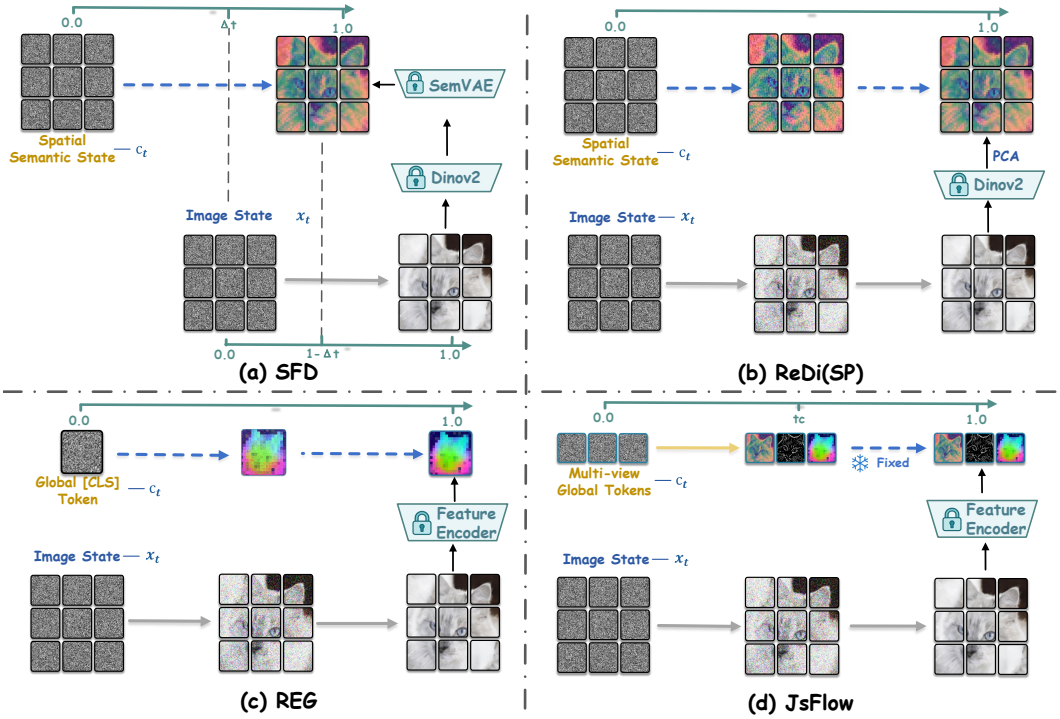


Figure 9: **Comparison of semantic-guided generation methods.** (a) SFD compresses DINOv2 patch features into spatial semantic latents using a separately pretrained SemVAE and adopts a three-stage schedule: semantic-only initialization, joint generation with a temporal offset Δt, and image-only refinement with fixed semantics. (b) ReDi jointly generates PCA-reduced DINOv2 patch features and image latents under a shared noise schedule. (c) REG uses a single global `[CLS]` token. (d) JsFlow jointly evolves image latents and multi-view global tokens, completing semantic generation at $t_c < 1$ and holding the tokens fixed during the remaining image transport.

Semantic representations. As shown in Fig. 9, SFD (Pan et al., 2025) compresses DINOv2 patch features with a separately trained SemVAE and concatenates the resulting spatial semantic latents

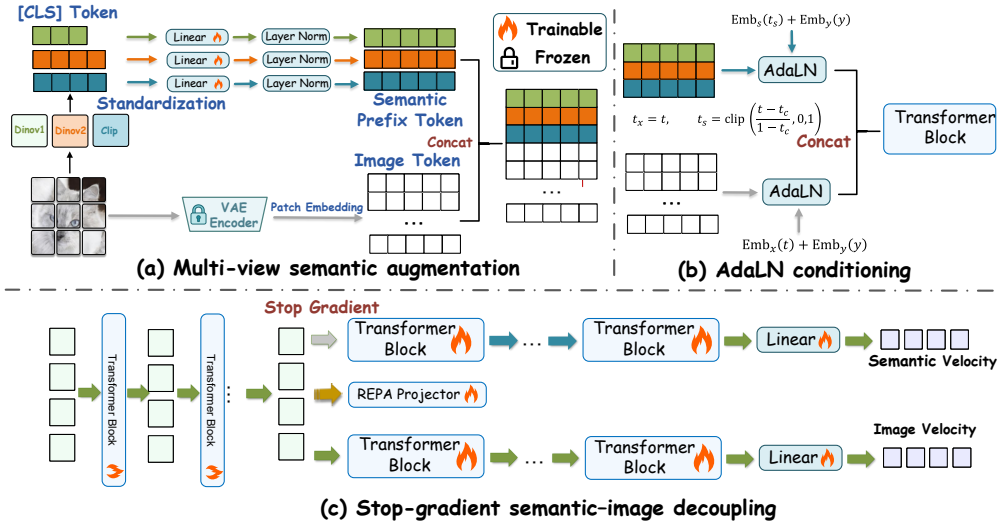


Figure 10: **Overview of the JsFlow design.** (a) Multi-view semantic augmentation using global class tokens from complementary frozen vision encoders. (b) Semantic-first asynchronous transport with group-wise AdaLN conditioning for semantic and image tokens. (c) Stop-gradient semantic–image decoupling in JsFlow†, where the semantic branch does not update the shared image backbone.

with image latents along the channel dimension. ReDi (Kouzelis et al., 2025) uses PCA-reduced DINOv2 patch features; its default implementation merges image and feature tokens, while Fig. 9(b) depicts its separate-token variant. REG (Wu et al., 2025) adds a single global class token to the image-token sequence. JsFlow instead adds three global tokens from CLIP, DINOv1, and DINOv2, obtaining complementary semantic views without a separately trained semantic autoencoder.

Generation schedules. REG and ReDi evolve image and semantic states under a shared schedule. Both SFD and JsFlow assign semantics a distinct schedule so that they reach their targets before image generation finishes. SFD uses a fixed temporal offset Δt, resulting in semantic-only initialization, joint evolution, and image-only refinement. JsFlow instead evolves the image throughout the trajectory: semantic tokens reach their targets at $t_c < 1$ and remain fixed during the remaining image transport, without an initial semantic-only phase. In our early-checkpoint comparison, SFD exhibits a lower IS than JsFlow. Its staggered evolution, particularly the semantic-only phase, may limit sample diversity.

C.2 JsFlow Design and Implementation

As illustrated in Fig. 10, JsFlow combines multi-view semantic augmentation with semantic-first asynchronous transport. We further enhance JsFlow with gradient-decoupled semantic prediction and denote this variant by JsFlow†.

Multi-view semantic augmentation. We augment the image state with multiple complementary semantic coordinates. Given an image $\mathbf{x}$, its global semantic representations are extracted as

$$\mathbf{c}_0^{(k)} = \mathrm{CLS}\left(\mathcal{E}_k(\mathcal{T}_k(\mathbf{x}))\right), \qquad k = 1, \ldots, K, \tag{84}$$

where $\mathcal{E}_k$ and $\mathcal{T}_k$ denote the k-th encoder and its image preprocessing operation, respectively. We use $K = 3$ frozen encoders: CLIP ViT-B, DINOv1 ViT-B, and DINOv2 ViT-B (Radford et al., 2021; Caron et al., 2021; Oquab et al., 2024). Their class-token dimensions are 512, 768, and 768, respectively. All semantic features are extracted offline.

Because features from different encoders have different coordinate scales, each semantic coordinate is standardized using training-set statistics:

$$\overline{\mathbf{c}}_t^{(k)} = \frac{\mathbf{c}_t^{(k)} - \boldsymbol{\mu}_k}{\max(\boldsymbol{\sigma}_k, \epsilon_{\text{sem}})}, \qquad \epsilon_{\text{sem}} = 10^{-2}. \tag{85}$$

The standardized features are mapped to the Transformer hidden dimension through independent trainable Linear–LayerNorm projections:

$$\mathbf{s}_t^{(k)} = \text{LN}_k\left(\mathbf{W}_k \overline{\mathbf{c}}_t^{(k)} + \mathbf{b}_k\right). \tag{86}$$

For LightningDiT-XL/1, the common hidden dimension is $H = 1152$.

In parallel, the frozen VA-VAE encodes the image into a $16 \times 16 \times 32$ latent tensor. With patch size one, patch embedding produces $N = 256$ image tokens. The semantic prefix and image tokens are concatenated along the sequence dimension:

$$\mathbf{Z}_t = [\mathbf{S}_t; \mathbf{X}_t] \in \mathbb{R}^{(K+N)\times H}, \qquad \mathbf{S}_t = \left[\mathbf{s}_t^{(1)}; \ldots; \mathbf{s}_t^{(K)}\right]. \tag{87}$$

These semantic coordinates complement the image state with information that is not already captured by the current image tokens.

Semantic-first asynchronous transport. We let semantic transport finish before image transport. Under the reverse clock used by our implementation, where $t = 1$ denotes noise and $t = 0$ denotes data, the image and semantic clocks are

$$t_x = t, \qquad t_s = \text{clip}\left(\frac{t - t_c}{1 - t_c}, 0, 1\right), \qquad t_c = 0.3. \tag{88}$$

The implementation value $t_c = 0.3$ corresponds to $t_c = 0.7$ under the generation-progress clock used in the main paper. The image and semantic states follow

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\boldsymbol{\epsilon}_x, \qquad \mathbf{c}_t^{(k)} = (1 - t_s)\mathbf{c}_0^{(k)} + t_s\boldsymbol{\epsilon}_c^{(k)}. \tag{89}$$

Consequently, for $t \le t_c$, the semantic coordinates have reached their clean endpoints and their target velocities are zero. This makes richer semantic information available throughout the remaining image transport.

As shown in Fig. 10(b), semantic and image tokens receive separate AdaLN conditions:

$$\mathbf{a}_s = \text{Emb}_s(t_s) + \text{Emb}_y(y), \qquad \mathbf{a}_x = \text{Emb}_x(t) + \text{Emb}_y(y). \tag{90}$$

The same AdaLN module maps these conditions to the shift, scale, and gate parameters used in each Transformer block. Semantic prefix tokens use $\mathbf{a}_s$, while image tokens use $\mathbf{a}_x$. Thus, the two token groups follow their respective transport clocks while sharing the attention and MLP backbone.

Gradient-decoupled semantic prediction. Standard JsFlow predicts image and semantic velocities with the shared network. To prevent semantic supervision from directly interfering with image optimization, JsFlow† predicts semantic velocities through a dedicated branch.

As shown in Fig. 10(c), the semantic branch is attached after the first $L_f = 8$ Transformer blocks. Let

$$\mathbf{H}_f = F_{1:L_f}(\mathbf{Z}_t) \tag{91}$$

denote the intermediate joint feature. Before passing this feature to the semantic branch, we apply a stop-gradient operation:

$$\widehat{\mathbf{u}}_C = \text{Head}_C\left(G_{1:L_s}\left(\text{sg}(\mathbf{H}_f)\right)\right), \qquad L_s = 4, \qquad \frac{\partial\, \text{sg}(\mathbf{H}_f)}{\partial \mathbf{H}_f} = \mathbf{0}. \tag{92}$$

The semantic tower and its prediction heads remain trainable, but the semantic loss does not update the shared image backbone. Meanwhile, the image branch continues through the remaining Transformer blocks to predict the image velocity.

Table 7: Architecture, JsFlow, optimization, and sampling configurations across different backbone scales.

Backbone	SiT-XL	LightningDiT-B	LightningDiT-L	LightningDiT-XL
Architecture				
#Params	675M	130M	458M	675M
Latent tokenizer	SD-VAE	VA-VAE	VA-VAE	VA-VAE
Latent input	$32 \times 32 \times 4$	$16 \times 16 \times 32$	$16 \times 16 \times 32$	$16 \times 16 \times 32$
Patch size	2	1	1	1
Layers	28	12	24	28
Hidden dimension	1152	768	1024	1152
Number of attention heads	16	12	16	16
JsFlow settings				
t_c	0.7	0.7	0.7	0.7
REPA visual encoder	DINOv2-B-reg	DINOv2-B-reg	DINOv2-B-reg	DINOv2-B-reg
REPA weight λ	0.5	0.5	0.5	0.5
REPA alignment depth	2	2	2	2
REPA similarity	cosine	cosine	cosine	cosine
Optimization				
Global batch size	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW
Weight decay	0	0	0	0
AdamW β_1	0.9	0.9	0.9	0.9
AdamW β_2 (80 epochs)	0.95	0.95	0.95	0.95
AdamW β_2 (800 epochs)	0.999	0.999	0.999	0.999
lr, epochs 1–40	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
lr, epochs 41–80	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}
lr, epochs 1–400	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
lr, epochs 401–800	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}
Sampling				
Sampling steps (before/after t_c)	175/75	175/75	175/75	175/75
Image CFG scale w_1	3.0	3.0	3.0	3.0
Semantic CFG scale w_2	1.25	1.25	1.25	1.25

Both JsFlow and JsFlow† use REPA to align intermediate image features with frozen DINOv2 patch features extracted from clean images (Yu et al., 2025). Unlike the semantic branch, the REPA path is not detached, so its supervision continues to optimize the shared backbone.

Wavelet and Fourier features. For the auxiliary comparison in Table 4, we replace the pretrained-encoder tokens with a single 128-dimensional deterministic semantic endpoint extracted from the normalized VA-VAE latent $x \in \mathbb{R}^{32\times16\times16}$. For Wavelet, we apply a two-level orthonormal Haar transform to each latent channel and collect the mean and centered log-RMS of the second-level low-frequency coefficients, together with the log-RMS of the level-1 and level-2 detail coefficients. For Fourier, we compute the spatial averages of $\sin(2^n\pi x)$ and $\cos(2^n\pi x)$ for $n \in \{7, 8\}$ in each channel. Both constructions produce $32 \times 4 = 128$ features. We apply a full-covariance whitening transform fitted on the training set and use the whitened vector as the semantic endpoint in the same JsFlow† transport and detached semantic branch. These features are extracted offline and require no pretrained visual encoder.

C.3 Training and Sampling Configurations

Training. We evaluate JsFlow with SiT-XL and three LightningDiT backbones (B/L/XL). All models are trained on ImageNet-1K at a resolution of 256×256. Images are center-cropped and resized before being encoded into the latent space. We use a frozen DINOv2-B-reg encoder to provide patch-level targets and apply REPA with weight $\lambda = 0.5$ at two alignment layers. The transition

point is $t_c = 0.7$ in the generation-progress clock used in this paper, which corresponds to 0.3 in the reverse clock used by the implementation.

All models are optimized with AdamW using a global batch size of 256, no weight decay, and $\beta_1 = 0.9$. For the 80-epoch setting, we use $\beta_2 = 0.95$, with a learning rate of 2×10^{-4} for the first 40 epochs and 2×10^{-6} afterward. For the 800-epoch setting, we use $\beta_2 = 0.999$, with a learning rate of 1×10^{-4} for the first 400 epochs and 2×10^{-6} for the remaining epochs. The complete settings are summarized in Table 7.

Sampling. We use the segmented JsFlow sampler with 250 steps: 175 steps before the semantic transition and 75 steps afterward. After the semantic state reaches its endpoint, it is kept fixed while the image state continues to evolve. We use EMA model weights and generate 50,000 samples for FID evaluation.

JsFlow predicts the image and semantic velocities jointly. Since these two components have different roles during generation, we apply separate CFG scales:

$$\begin{aligned} \widetilde{\mathbf{v}}_t^{\mathrm{img}} &= \mathbf{v}_{u,t}^{\mathrm{img}} + w_1 \left(\mathbf{v}_{c,t}^{\mathrm{img}} - \mathbf{v}_{u,t}^{\mathrm{img}} \right), \\ \widetilde{\mathbf{v}}_t^{\mathrm{sem}} &= \mathbf{v}_{u,t}^{\mathrm{sem}} + w_2 \left(\mathbf{v}_{c,t}^{\mathrm{sem}} - \mathbf{v}_{u,t}^{\mathrm{sem}} \right), \end{aligned} \tag{93}$$

where c and u denote the class-conditional and unconditional predictions, respectively. We set the image guidance scale to $w_1 = 3.0$ and the semantic guidance scale to $w_2 = 1.25$. The smaller semantic guidance scale prevents over-guidance of the semantic trajectory while retaining sufficiently strong image-level class guidance.

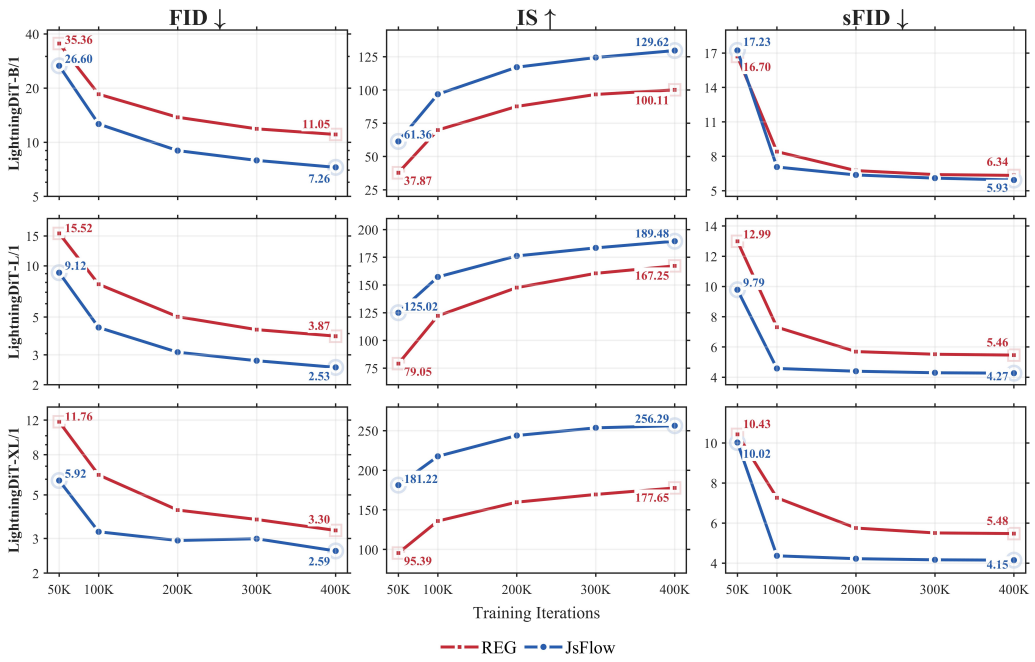


Figure 11: **Training convergence of REG and JsFlow across LightningDiT scales on ImageNet 256×256 without guidance.** Rows correspond to LightningDiT-B/1, L/1, and XL/1, while columns report FID, IS, and sFID. JsFlow consistently improves FID and IS and achieves better sFID after the early stage of training.

D Additional Experiments and Ablations

D.1 Extended Quantitative Results

Training Convergence across Model Scales. We compare REG and JsFlow using the same LightningDiT-B/1, L/1, and XL/1 backbones on ImageNet 256×256. To isolate the effect of the training objective, all results are reported without classifier-free guidance. Figure 11 shows FID, IS, and sFID from 50K to 400K training iterations. JsFlow consistently obtains lower FID and higher IS across all three model scales and improves substantially faster during early training. In particular, JsFlow reaches a lower FID than the corresponding REG model at 400K using only 200K iterations for B/1 and L/1, and 100K iterations for XL/1.

The advantage remains clear at 400K iterations. For B/1, JsFlow reduces FID from 11.05 to 7.26 and increases IS from 100.11 to 129.62. For L/1, FID decreases from 3.87 to 2.53, while IS increases from 167.25 to 189.48. For XL/1, JsFlow obtains an FID of 2.59 and an IS of 256.29, compared with 3.30 and 177.65 for REG. JsFlow also reduces the final sFID from 6.34 to 5.93, from 5.47 to 4.27, and from 5.48 to 4.15 for B/1, L/1, and XL/1, respectively. Detailed results at every checkpoint are provided in Table 8.

Overall, the gains are consistent across model capacities rather than being specific to a single backbone size. The stronger early-checkpoint results also indicate that JsFlow improves training efficiency in addition to final generation quality.

D.2 Semantic Representation and Transport Schedule Ablations

Semantic Encoder Ablation without REPA. The preceding experiments use REPA together with JsFlow, so the effect of the semantic representation may be coupled with representation alignment. To isolate the semantic coordinates themselves, we remove REPA and classifier-free guidance and compare different frozen visual encoders under the same training setup. All variants use SiT-B/2, are

Table 8: **Detailed comparison of REG and JsFlow on ImageNet** 256×256 **without guidance.** Both methods use the same LightningDiT-B/1, L/1, and XL/1 backbone architectures.

(a) REG

Backbone	Iter.	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
LightningDiT-B/1	50K	35.36	16.71	37.87	0.55	0.51
LightningDiT-B/1	100K	18.50	8.41	69.92	0.66	0.58
LightningDiT-B/1	200K	13.78	6.75	87.65	0.69	0.60
LightningDiT-B/1	300K	11.88	6.41	96.66	0.70	0.59
LightningDiT-B/1	400K	11.05	6.34	100.11	0.70	0.60
LightningDiT-L/1	50K	15.52	12.99	79.05	0.72	0.51
LightningDiT-L/1	100K	7.80	7.31	122.24	0.76	0.56
LightningDiT-L/1	200K	5.03	5.70	147.65	0.77	0.59
LightningDiT-L/1	300K	4.22	5.52	160.52	0.78	0.60
LightningDiT-L/1	400K	3.87	5.47	167.25	0.78	0.60
LightningDiT-XL/1	50K	11.76	10.43	95.39	0.75	0.52
LightningDiT-XL/1	100K	6.32	7.26	135.85	0.78	0.55
LightningDiT-XL/1	200K	4.19	5.75	159.72	0.79	0.59
LightningDiT-XL/1	300K	3.75	5.51	169.46	0.78	0.60
LightningDiT-XL/1	400K	3.30	5.48	177.65	0.78	0.61

(b) JsFlow

Backbone	Iter.	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
LightningDiT-B/1	50K	26.60	17.23	61.36	0.61	0.50
LightningDiT-B/1	100K	12.63	7.07	96.81	0.70	0.56
LightningDiT-B/1	200K	8.99	6.38	117.14	0.72	0.58
LightningDiT-B/1	300K	7.93	6.10	124.41	0.73	0.58
LightningDiT-B/1	400K	7.26	5.93	129.62	0.74	0.59
LightningDiT-L/1	50K	9.12	9.79	125.02	0.77	0.51
LightningDiT-L/1	100K	4.34	4.58	157.26	0.79	0.57
LightningDiT-L/1	200K	3.11	4.40	176.22	0.80	0.58
LightningDiT-L/1	300K	2.77	4.30	183.48	0.80	0.59
LightningDiT-L/1	400K	2.53	4.27	189.48	0.80	0.60
LightningDiT-XL/1	50K	5.91	10.02	181.22	0.84	0.45
LightningDiT-XL/1	100K	3.25	4.37	217.62	0.85	0.53
LightningDiT-XL/1	200K	2.93	4.22	243.92	0.85	0.54
LightningDiT-XL/1	300K	2.99	4.17	253.63	0.86	0.54
LightningDiT-XL/1	400K	2.59	4.15	256.29	0.84	0.56

Table 9: **Semantic encoder ablation without REPA.** All JsFlow variants use SiT-B/2, are trained for 600K iterations without REPA or guidance, and use $t_c = 0.5$.

Representation	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Multi-Enc	**13.20**	7.03	**101.88**	**0.74**	0.50
DINOv1 ViT-B	13.31	**6.45**	97.90	0.70	0.55
DINOv2 ViT-B	16.46	6.58	94.17	0.66	**0.58**
CLIP ViT-B	19.78	6.67	77.41	0.66	0.55
MAE ViT-L	55.39	19.39	37.16	0.53	0.53
MoCo-v3 ViT-B	117.14	33.70	13.94	0.35	0.28

trained for 600K iterations, and use $t_c = 0.5$. Table 9 therefore measures the effect of the semantic representation without the additional benefit of REPA.

Multi-Enc combines CLIP, DINOv1, and DINOv2 representations. It achieves the best FID of 13.20, the highest IS of 101.88, and the highest precision of 0.74. Among the individual encoders, DINOv1 performs best in FID and sFID, reaching 13.31 and 6.45, respectively. DINOv2 obtains the highest recall of 0.58, indicating better sample coverage but lower fidelity than DINOv1. CLIP remains competitive but is weaker on FID and IS. In contrast, MAE and MoCo-v3 perform substantially worse, even though MAE uses a larger ViT-L encoder.

These results show that semantic representation remains important even when REPA is completely removed. In particular, the performance differences between encoders cannot be attributed solely to the REPA alignment loss. Instead, generation quality depends directly on how informative and complementary the semantic coordinates are. The strong performance of Multi-Enc further suggests that combining multiple semantic views is more effective than simply increasing the capacity of a single encoder.

Image-loss Analysis. We further compare the optimization behavior of JsFlow and JsFlow† across five individual semantic encoders. JsFlow† uses gradient-decoupled semantic prediction to reduce interference between semantic prediction and image generation. Figure 12 shows their image-loss trajectories, with the final 200K iterations enlarged in each inset.

JsFlow† yields a lower average image training loss for all five encoders. The largest reductions occur with DINOv1 and DINOv2, for which the full-trajectory loss decreases by 0.576% and 0.540%, respectively. The reductions are smaller for CLIP, MAE, and MoCo-v3, suggesting that the benefit of optimization decoupling depends on the semantic representation. Together with Table 9, these results show that informative and complementary semantic features improve generation, while decoupling the semantic objective can further reduce interference with image optimization. Since

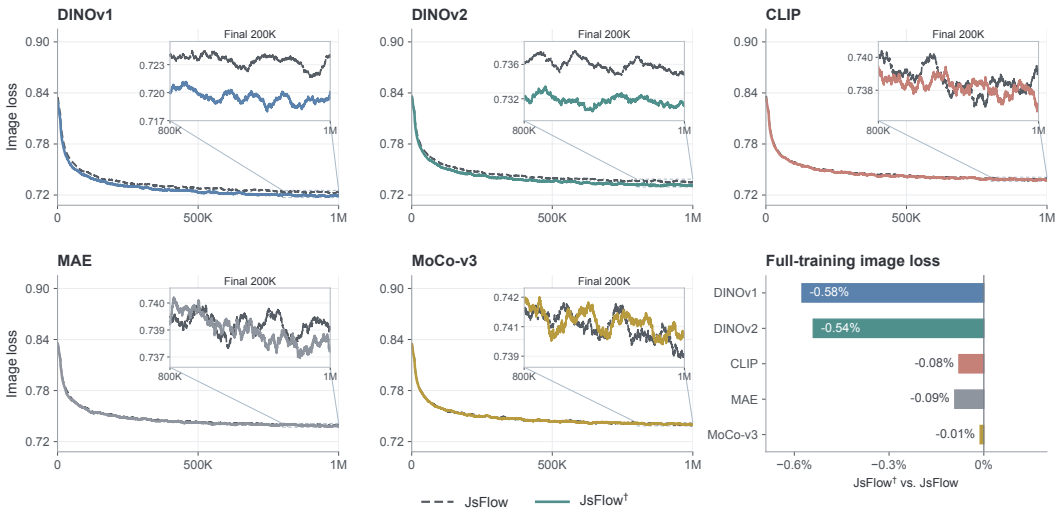


Figure 12: **Image training loss of JsFlow and JsFlow† across five semantic encoders.** Each inset enlarges the final 200K training steps. The rightmost panel reports the relative change in image loss averaged over the full training trajectory, where negative values indicate lower loss for JsFlow†.

Table 10: Semantic transport schedules and ablation results ($N = 1{,}500$).

Schedule	Semantic displacement $\beta(t)$	Continuity at t_c	FID ↓	Δ
Hard	$\begin{cases} t/T, & t < T, \\ 1, & t \geq T \end{cases}$	C^0	**32.662**	—
Global-soft	$\begin{cases} g(u), & t < T, \\ 1, & t \geq T \end{cases}$	C^2	34.930	+2.269
Local-C^1	$\begin{cases} t/T, & t \leq T-\delta, \\ 1-\frac{\delta}{T}x^2(2-x), & T-\delta < t < T, \\ 1, & t \geq T \end{cases}$	C^1	34.802	+2.140

training loss is not itself a generation metric, we use it as evidence about optimization behavior rather than as a substitute for FID or sFID.

Transport Schedule Choices. We compare three ways of transporting the semantic representation: Hard, Global-soft, and Local-C^1. We use generation progress $t \in [0, 1]$, from noise to data, and set the transition point to $T \equiv t_c = 0.7$. For the local smoothing interval, we use $\delta = 0.05$. We further define

$$u = \frac{t}{T}, \qquad x = \frac{T-t}{\delta}, \qquad g(u) = u + 4u^3 - 7u^4 + 3u^5. \tag{94}$$

Hard uses a linear ramp before t_c and remains constant afterward. Global-soft replaces the complete ramp with a polynomial whose first two derivatives vanish at t_c. Local-C^1 keeps the linear Hard target over most of the trajectory and smooths only the short interval $[T-\delta, T]$. Their exact expressions and quantitative results are listed in Table 10; the corresponding displacement, velocity, and acceleration curves are shown in Figure 13.

Although the Hard schedule has a discontinuous first derivative at t_c, it achieves the best FID of 32.662. Global-soft and Local-C^1 improve the smoothness of the transition, but obtain FIDs of 34.930 and 34.802, respectively. One possible explanation is that Hard provides a simpler supervision target: before t_c, its displacement is linear and its velocity is constant. In contrast, the smoothed schedules introduce nonlinear targets and larger local derivative variations, which may make optimization more difficult under the same training budget. These results suggest that increased schedule smoothness does not necessarily lead to better generation quality.

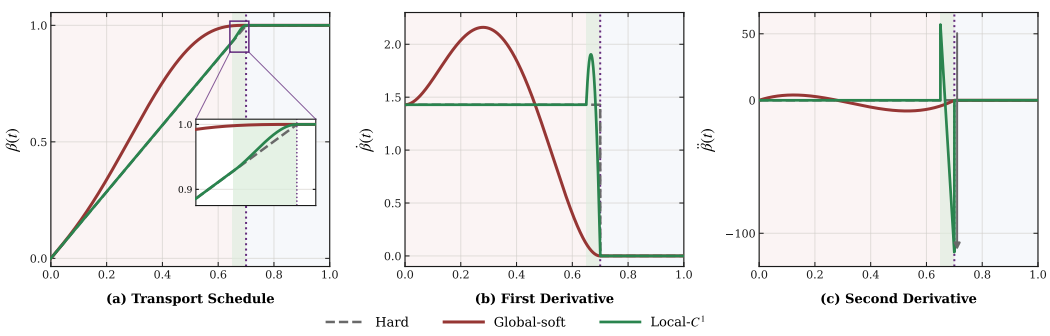


Figure 13: Displacement and its first two derivatives for the three transport schedules. The inset in **(a)** shows the local transition near $t_c = 0.7$.

D.3 Sampling Steps and Guidance Ablations

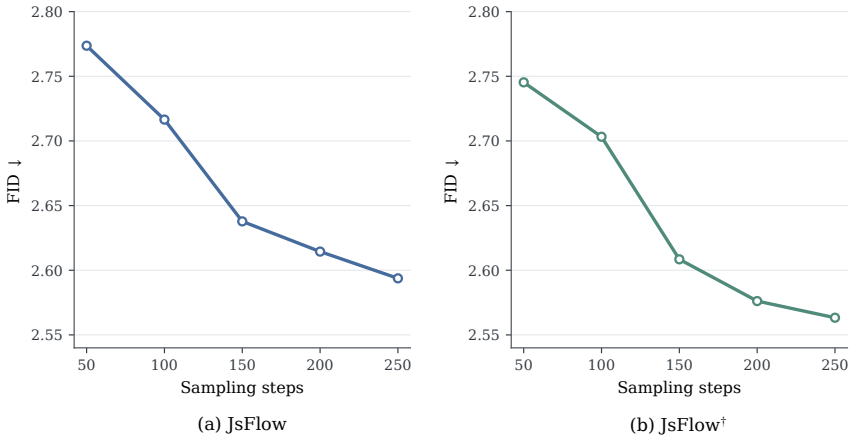


Figure 14: Effect of the number of sampling steps on generation quality for **(a) JsFlow** and **(b) JsFlow**†. FID is computed using 50K generated samples without classifier-free guidance. Both models exhibit a consistent reduction in FID as the sampling budget increases from 50 to 250 steps.

Sampling Steps. Figure 14 studies the effect of the sampling budget on the generation quality of JsFlow and JsFlow†. For both models, FID decreases monotonically as the number of sampling steps increases, indicating that the learned joint image–semantic velocity field remains stable under increasingly accurate numerical integration. Specifically, the FID of JsFlow decreases from 2.77 at 50 steps to 2.59 at 250 steps, while that of JsFlow† decreases from 2.75 to 2.56. The largest improvement is obtained when increasing the sampling budget from 100 to 150 steps, after which the gain gradually saturates. JsFlow† consistently achieves a lower FID across all evaluated sampling budgets, with a final FID of 2.56 at 250 steps. These results show that both variants already produce competitive samples with a relatively small number of function evaluations, while additional sampling steps provide further, though diminishing, improvements in generation quality.

CFG Configuration. Table 11 reports FID-5K for 16 representative CFG configurations. All configurations are evaluated with the same JsFlow-XL model trained for 400K steps, while all other sampling settings are kept fixed.

The scale sweep includes the default $w_1 = w_2 = 3.0$ configuration together with representative symmetric and asymmetric alternatives. The interval sweep shows that applying guidance throughout the full trajectory degrades FID, while ending guidance before the final refinement stage performs better.

Table 11: Component-wise CFG ablations using the same JsFlow-XL model trained for 400K steps. Left: image and semantic guidance scales with interval $[0, 0.90]$. Right: guidance interval configurations.

(a) Guidance scales

w_1 (image)	w_2 (semantic)	FID-5K ↓
1.00	1.00	8.828
1.50	1.50	8.409
2.75	2.75	8.600
3.00	**3.00**	8.693
3.50	3.50	8.897
4.00	4.00	9.075
2.75	1.125	8.330
3.00	1.25	**8.325**

(b) Guidance intervals

w_1	w_2	Interval	FID-5K ↓
2.75	1.70	$[0, 0.800]$	8.525
2.75	1.70	$[0, 0.825]$	8.480
2.75	1.70	$[0, 0.850]$	8.439
2.75	1.70	$[0, 0.900]$	8.344
3.00	1.00	$[0, 0.900]$	**8.339**
3.00	1.00	$[0.10, 0.90]$	8.342
3.00	1.00	$[0, 0.950]$	9.382
3.00	1.00	$[0, 1.000]$	12.601

E More Visualization Results

In this section, we provide additional qualitative results. We first visualize how the images generated by different methods evolve over the course of training, illustrating their convergence behavior and the gradual emergence of recognizable semantic structures. We then present additional class-conditional samples from selected categories under both guided and unguided sampling settings, providing a broader qualitative assessment of generation quality, semantic consistency, and sample diversity.

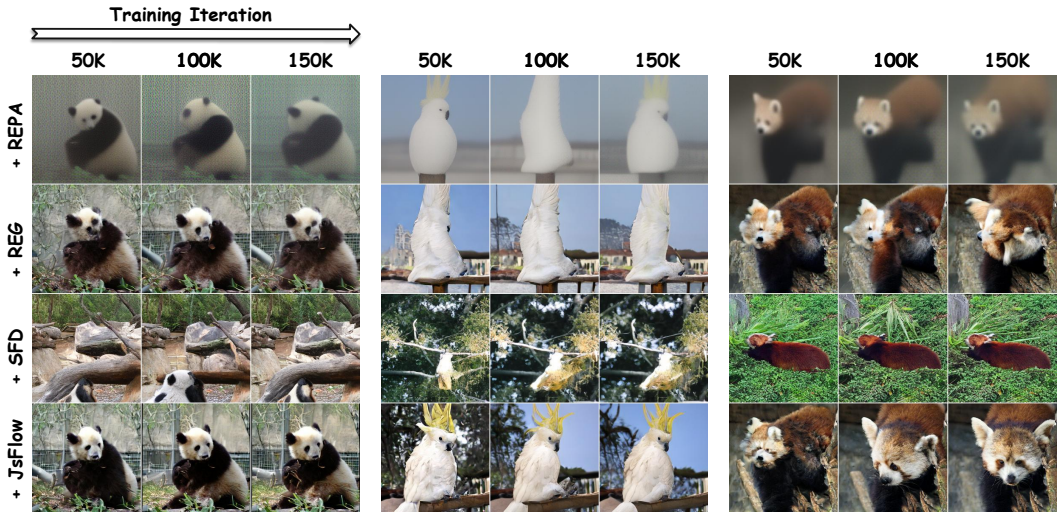


Figure 15: **Qualitative comparison across training iterations.** Generated samples from REPA, REG, SFD, and JsFlow at 50K, 100K, and 150K training iterations. JsFlow forms recognizable and consistent semantic structures earlier than the compared methods.



Figure 16: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Bald eagle (22)”.



Figure 17: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Red panda (387)”.

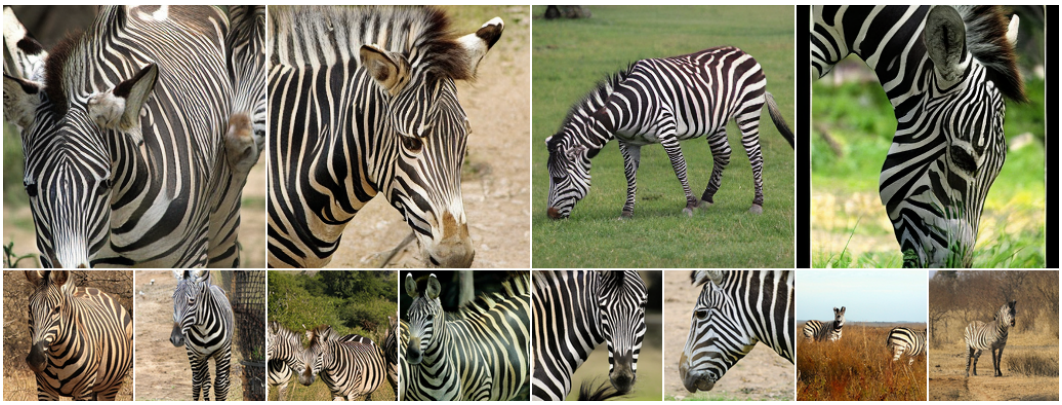


Figure 18: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Zebra (340)”.



Figure 19: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Pineapple (953)”.



Figure 20: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Agaric mushroom (992)”.



Figure 21: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Hot-air balloon (417)”.



Figure 22: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Volcano (980)”.



Figure 23: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Castle (483)”.



Figure 24: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, using classifier-free guidance with $w = 3.0$. All images are generated with the ImageNet class label “Coral reef (973)”.



Figure 25: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "Flamingo (130)".



Figure 26: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "King penguin (145)".

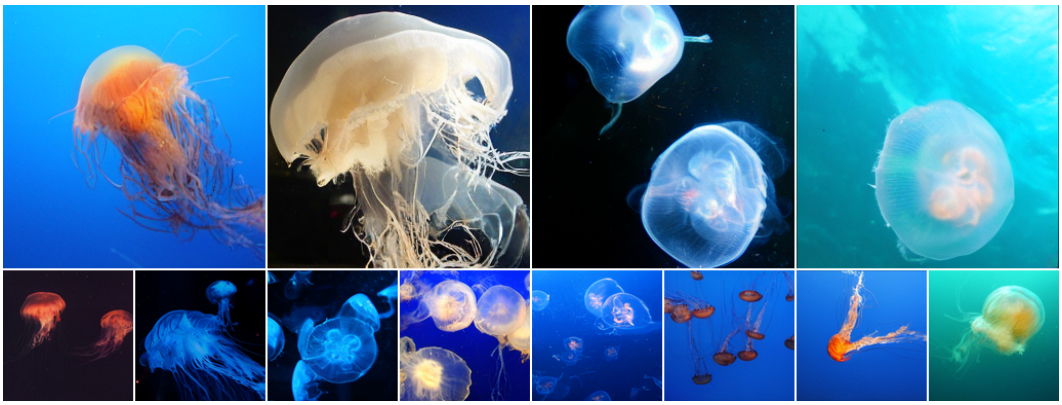


Figure 27: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "Jellyfish (107)".



Figure 28: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "Giant panda (388)".



Figure 29: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "Orange (950)".



Figure 30: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label "Cheeseburger (933)".



Figure 31: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label “Steam locomotive (820)”.



Figure 32: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label “Lighthouse (437)”.



Figure 33: Selected samples from JsFlow with the LightningDiT-XL/1 backbone after 4M training iterations, without classifier-free guidance ($w = 1.0$). All images are generated with the ImageNet class label “Airliner (404)”.