Better Source, Better Flow: Learning Condition-Dependent Source Distribution for Flow Matching

Junwan Kim[1]* **Jiho Park**[2]* **Seonghu Jeon**[2] **Seungryong Kim**[2]†

[1]New York University [2]KAIST AI

https://junwankimm.github.io/CSFM

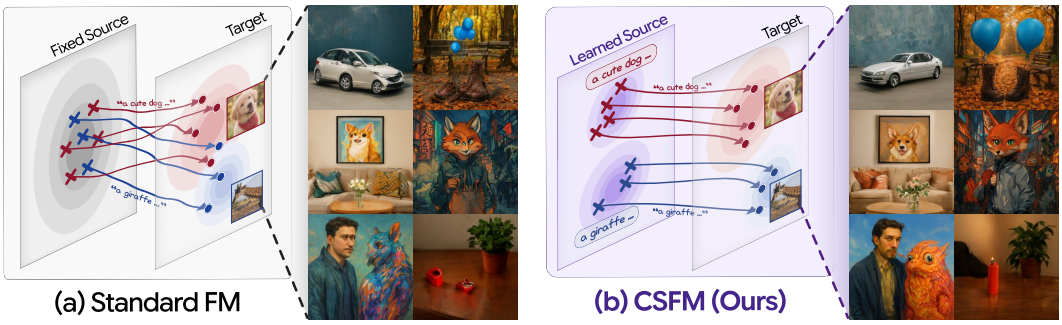

Figure 1: **Condition-dependent Source Flow Matching (CSFM).** Flow matching does not require the source distribution to be a fixed standard Gaussian. We leverage this flexibility by learning a condition-dependent source distribution, making flow matching easier to train and improving generation performance. Qualitative examples illustrate improved generation quality.

Abstract

Flow matching has recently emerged as a promising alternative to diffusion-based generative models, particularly for text-to-image generation. Although flow matching places no restriction on the source distribution, most existing systems still inherit a standard Gaussian from diffusion models, and the source is rarely treated as an optimization target at this scale. Recent works have begun to revisit this choice through condition-dependent or learned sources, yet evidence that such designs are effective in modern text-to-image systems—with high-dimensional latents and tightly integrated conditioning—remains limited. In this work, we study condition-dependent source distributions for flow matching along three axes: *why* source learning helps, through the lens of the intrinsic variance term in the flow-matching objective; *how* to make it work in modern text-to-image systems, where variance-only regularization and directional source–target alignment are critical for stable end-to-end training; and *when* it is most beneficial, by connecting source design to recent representational advances in generative modeling and identifying target representation regimes in which learning the source yields the largest gains. Extensive experiments across multiple text-to-image benchmarks, architectures, and scales demonstrate that principled source design yields consistent and robust improvements—including up to $\mathbf{3.01}\times$ faster convergence in FID and $\mathbf{2.48}\times$ in CLIP Score—and outperforms representative condition-dependent source and condition-aware coupling methods.

*Equal contribution.

†Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Flow Matching [38, 41] has recently emerged as a powerful framework for generative modeling, demonstrating strong performance across many domains. In particular, text-to-image (T2I) systems show that *conditional* flow matching[3]can generate high-fidelity images with strong prompt adherence [17, 33, 44, 48], suggesting that it is a promising alternative to diffusion-based models for conditional image generation.

At its core, flow matching learns a continuous-time velocity field that transports samples from a *source* distribution to a *target* distribution. Unlike diffusion models, which rely on stochastic noise injection [25, 56] or denoising score matching [57, 58], flow matching directly models deterministic dynamics. More importantly, it places no restriction on the source distribution. This flexibility is especially appealing for conditional generation, since the conditioning signal can be used not only to modulate the velocity network, but also to define a condition-dependent source distribution.

Despite this conceptual advantage, most existing flow matching methods still adopt a standard Gaussian source, which carries no information about the target distribution. This choice is largely inherited from diffusion models and motivated by simplicity rather than by the structure of the conditional generation problem. Recent work has begun to revisit this default from two complementary directions. One line improves the coupling between a Gaussian source and the data, often through optimal transport [13, 47, 60]; another modifies the source itself by constructing or learning a condition-dependent source [9, 21, 28, 40].

Together, these works show that the source distribution can encode useful structure. However, for modern conditional flow matching, especially text-to-image generation, a systematic understanding remains incomplete. OT-based couplings clarify the role of transport ambiguity, but their benefits are difficult to realize in high-dimensional conditional regimes and may not transfer when applied naively [13]. Existing condition-dependent source designs demonstrate that conditioning can be injected into the source, but they offer limited guidance on *why* a learned source reduces the difficulty of the flow-matching objective, *how* to make source learning consistently beneficial in modern T2I regimes with high-dimensional latents [17, 33, 73], large-scale datasets [11, 50, 52], and strong conditioning [10, 12, 48], or *when* its benefits should be most pronounced.

In this work, we study condition-dependent source learning as a first-class design problem for conditional flow matching. We propose *Condition-dependent Source Flow Matching* (CSFM), a framework that jointly learns a condition-dependent source distribution and the velocity field under the flow-matching objective. Using modern text-to-image generation as a high-dimensional and strongly conditioned testbed, we address three central questions: *why* source learning improves flow matching, *how* it can be made stable and effective in practice, and *when* its benefits are most pronounced. Our main contributions are:

- **Why source learning helps.** We connect jointly learned condition-dependent sources to the intrinsic variance term of the flow-matching objective, which characterizes ambiguity in velocity supervision. Empirically, CSFM provides cleaner supervision and improves optimization through lower gradient variance, straighter transport paths, and faster convergence.
- **How to make source learning effective in modern T2I.** We identify key obstacles to stable source learning, including source collapse during joint optimization and weak supervision to the source generator in strongly conditioned T2I architectures. We show that variance-only regularization with directional source–target alignment enables stable end-to-end training without constraining source mobility.
- **When source learning is most beneficial.** We show that the effectiveness of condition-dependent sources depends on the geometry of the target representation space. In particular, source learning yields larger gains in semantically structured target spaces, linking source design to recent advances in representations for generative modeling [10, 55, 71, 73].
- **Empirical validation at modern T2I scale.** Through extensive experiments and ablations across multiple text-to-image benchmarks, CSFM consistently improves over Gaussian-source baselines, achieving up to $\mathbf{3.01}\times$ faster FID convergence and $\mathbf{2.48}\times$ faster CLIP Score convergence, and outperforms representative condition-dependent source and coupling methods.

[3]In this paper, *conditional* flow matching refers to flow matching conditioned on an external variable. This should not be confused with Conditional Flow Matching (CFM) [38], where "conditional" instead refers to conditioning on X_1.

2 Related work

We discuss the most relevant related work here and provide additional details in Appx. K.

Recent works have revisited the standard Gaussian source in flow matching from two directions. The first modifies the source–target coupling while keeping the Gaussian source fixed. OT-based methods reduce path ambiguity through improved pairings [47, 60], and C^2OT extends this idea to conditional generation through condition-aware coupling [13]. However, these benefits are difficult to realize in high-dimensional conditional settings, where reliable minibatch OT estimation becomes challenging.

The second direction modifies the source distribution itself. CPD [28] constructs a condition-dependent prior through a separate regression objective decoupled from the FM loss, CAR-Flow [9] learns condition-dependent reparameterizations to align source and target distributions in simpler class-conditional settings, and CrossFlow and FlowTok [21, 40] demonstrate feasible text-dependent sources at T2I scale, but do not analyze source learning through FM training dynamics. Together, these works show that the source can encode conditioning information, but leave open a systematic understanding of *why* source learning helps under the FM objective, *how* to make source learning consistently beneficial beyond limited-scale T2I settings, and *when* its benefits should be substantial.

3 Preliminaries

3.1 Flow Matching

Flow Matching (FM) defines a generative process by solving an ordinary differential equation (ODE):

$$\frac{d}{dt}X_t = v_\theta(X_t, t), \quad t \in [0, 1], \tag{1}$$

where $v_\theta(\cdot, t)$ is a neural vector field parameterized by θ. The goal of FM is to find θ such that the push-forward of a source distribution p_0 through the ODE matches a target distribution p_1.

Typically, we consider a probability path p_t and an associated marginal velocity field u_t that generates p_t. For a given coupling $(X_0, X_1) \sim \pi$ with $X_0 \sim p_0$ and $X_1 \sim p_1$, we define the velocity target as:

$$\Delta := X_1 - X_0. \tag{2}$$

Assuming a linear interpolation path $X_t = (1-t)X_0 + tX_1$, the marginal velocity field at a point x and time t is expressed as

$$u_t(x) = \mathbb{E}[\Delta \mid X_t = x]. \tag{3}$$

The FM objective minimizes the mean squared error between the neural vector field and the velocity target:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,\pi(X_0,X_1)}\left[||v_\theta(X_t, t) - \Delta||^2\right], \tag{4}$$

where $t \sim p(t)$ is a probability distribution over $[0, 1]$. While each conditional path $t \mapsto X_t$ is linear, the learned marginal vector field generally induces curved trajectories because it aggregates multiple, potentially conflicting velocities Δ at the same spacetime point (x, t).

For conditional flow matching, the target X_1 is paired with condition C, and the velocity field additionally takes C as input. With the standard Gaussian source, one typically uses the condition-agnostic coupling $\pi(X_0, X_1, C) = p_0(X_0)p_1(X_1, C)$:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,\pi(X_0,X_1,C)}\left[||v_\theta(X_t, t, C) - \Delta||^2\right]. \tag{5}$$

3.2 Bias-Variance decomposition of Flow Matching loss

To analyze learning dynamics, the FM objective in Eq. (4) decomposes into two components (see Appx. C):

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \underbrace{\mathbb{E}_{t,\pi(X_0,X_1)}\Big[\|v_\theta(X_t, t) - u_t(X_t)\|^2\Big]}_{\text{Approximation Error}} + \underbrace{\mathbb{E}_{t,\pi(X_0,X_1)}\Big[\mathrm{Var}(\Delta \mid X_t)\Big]}_{\text{Intrinsic Variance}}. \tag{6}$$

The Approximation Error measures how well the model recovers the marginal velocity field u_t, while the Intrinsic Variance is an irreducible term determined by the coupling π. For linear interpolants,

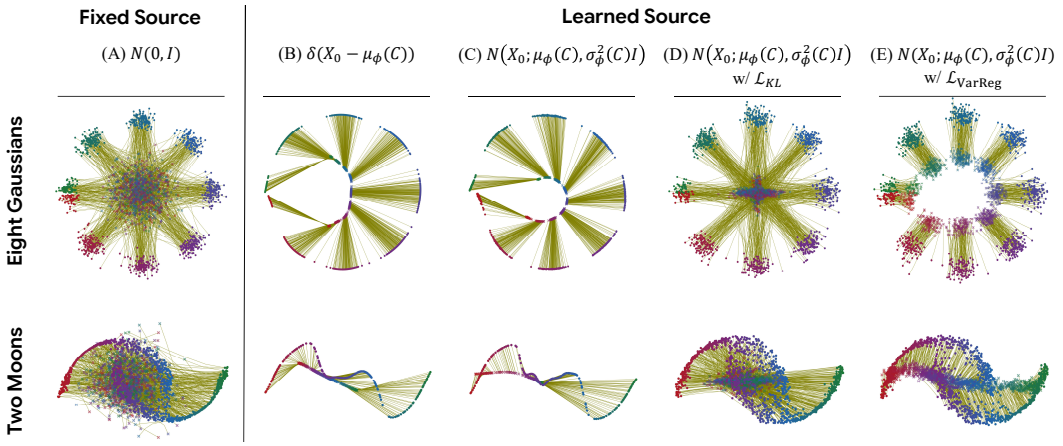

Figure 2: **Analysis of CSFM designs.** We investigate the effect of the source designs using two two-dimensional synthetic datasets with continuous conditions: Eight Gaussians with polar angle condition and Two Moons with x-coordinate condition. We visualize the transport trajectories, where '$\times$' denotes source points X_0 and '$\bullet$' denotes points X_1^{sampled} generated by the flow model. Colors indicate the conditioning variable. **(A) Fixed Standard Gaussian:** Independent coupling results in entangled paths and high intrinsic variance. **(B) Deterministic Mapping:** The flow model with a deterministically mapped source fails to reconstruct the original target distribution. **(C) Conditional Gaussian:** Although the source is modeled as a condition-dependent Gaussian, its variance collapses during training, resulting in insufficient support and an inability to recover the target distribution. **(D) Conditional Gaussian with Standard KL regularization:** While preventing collapse to a deterministic mapping, the constraint on $\mu_\phi(C)$ limits the mobility of the source, yielding entangled trajectories. **(E) Conditional Gaussian with VarReg:** Variance-only regularization prevents collapse while allowing the conditional mode $\mu_\phi(C)$ to move, resulting in a target-aligned source distribution and disentangled trajectories.

this variance characterizes path ambiguity: it vanishes when Δ is uniquely determined by X_t, or equivalently when interpolants do not intersect [43]. Thus, high intrinsic variance indicates that multiple trajectories crossing the same spacetime point (x, t) provide conflicting velocity supervision.

Prior works show that reducing intrinsic variance improves coupling quality, leading to faster convergence and more stable flow learning [47, 60]. OT-based methods affect this term by explicitly changing sample-level source–target pairings. However, explicit OT matching becomes unreliable in high-dimensional conditional settings, limiting their practical effectiveness [13].

4 Method

In this section, we present *Condition-dependent Source Flow Matching* (CSFM), which jointly learns a source distribution and the velocity field under the flow-matching objective (Fig. 1). We formulate source learning as adaptive source–target coupling, then introduce two components for stable training: variance-only regularization to prevent source collapse while preserving source mobility, and directional source–target alignment to improve source learning under complex conditioning.

4.1 Learning condition-dependent sources under flow matching objective

In conditional generation scenarios such as text-to-image generation, the conditioning variable C is naturally paired with the data random variable X_1. We leverage this relationship by replacing the condition-agnostic Gaussian source in Eq. (5) with a learnable condition-dependent source distribution $p_\phi(X_0 \mid C)$. This defines a learnable condition-mediated coupling $\pi_\phi(X_0, X_1, C) = p_\phi(X_0 \mid C)p_1(X_1, C)$, where the source marginal adapts to the condition.

Specifically, we introduce a source generator $g_\phi(\cdot)$ that maps the condition C to the parameters of $p_\phi(X_0 \mid C)$. We sample $X_0 \sim p_\phi(X_0 \mid C)$ and jointly train the flow model $v_\theta(\cdot)$ and the source generator $g_\phi(\cdot)$ under the conditional FM loss:

$$\begin{aligned} &\mathcal{L}_{\text{FM}}(\theta, \phi) = \mathbb{E}_{t, \pi_\phi(X_0, X_1, C)} \left[\| v_\theta(X_t, t, C) - \Delta \|^2 \right], \\ &\text{where } X_0 \sim p_\phi(X_0 \mid C), \quad \Delta = X_1 - X_0, \quad X_t = (1-t)X_0 + tX_1. \end{aligned} \tag{7}$$

By making the source–target coupling learnable through $p_\phi(X_0 \mid C)$, CSFM exposes the induced coupling to the FM objective rather than treating it as fixed. Since intrinsic variance is coupling-dependent (Eq. 6), source learning can reduce ambiguity in the velocity target, as verified in toy settings in Appx. A.2. This adaptive coupling yields fewer path intersections and cleaner velocity supervision, manifesting as lower gradient variance, faster convergence, and straighter flows. We visualize this effect in Fig. 2 and provide quantitative evidence in Sec. 5.2.

4.2 Preventing source collapse while preserving source mobility

While a learnable conditional source can reduce velocity ambiguity, careful design is required.

Conditional Gaussian for sufficient support. A straightforward source design is a deterministic mapping, $p_\phi(X_0 \mid C) = \delta(X_0 - \mu_\phi(C))$. However, this choice severely restricts the support[4] of the source X_0. As established by [34], an overly concentrated source causes path entanglement and degrades flow matching performance. We empirically demonstrate this failure mode for deterministic conditional sources in toy experiments (Fig. 2 (B)), where the flow model fails to recover the distribution of X_1. To address this, we instantiate $p_\phi(X_0 \mid C)$ as a conditional Gaussian:

$$p_\phi(X_0 \mid C) = \mathcal{N}\big(\mu_\phi(C),\, \mathrm{diag}(\sigma_\phi^2(C))\big)\,, \tag{8}$$

which provides full support when $\sigma_\phi^2(C) > 0$ and allows reparameterizable sampling [31].

Variance regularization for collapse prevention. A learnable Gaussian source can adapt both its conditional location and stochastic support, but joint training with the flow model often drives the source variance $\sigma_\phi^2(C)$ toward zero (Fig. 2 (C)), since shrinking the source variance directly reduces the intrinsic variance in Eq. (6). To counteract this effect, a common strategy for Gaussian parameterization is to regularize the distribution toward a standard normal prior [21, 40]. However, such regularization constrains both the variance and mean, forcing $\mu_\phi(C)$ toward the origin, which we find unnecessarily restricts the flexibility of the source distribution and degrades performance. As illustrated in Fig. 2 (D), this constraint prevents the source from relocating toward target modes, resulting in entangled transport paths. To avoid this limitation, we adopt a variance-only regularization that penalizes deviations of $\sigma_\phi^2(C)$ from unit variance while leaving the mean $\mu_\phi(C)$ unconstrained:

$$\mathcal{L}_{\mathrm{VarReg}}(\phi) = \mathbb{E}_C\Big[D_{\mathrm{KL}}\big(\mathcal{N}(\mu_\phi(C), \mathrm{diag}(\sigma_\phi^2(C)))\|\mathcal{N}(\mu_\phi(C), \mathbf{I})\big)\Big] \tag{9}$$

By leaving $\mu_\phi(C)$ unconstrained while anchoring the variance, the source distribution can relocate toward condition-compatible target regions without collapsing its support (Fig. 2 (E)). This avoids the mean-constraining effect of standard KL regularization and yields simpler, less entangled transport geometry with cleaner supervision signals.

4.3 Strengthening source supervision under complex conditioning

Unlike simplified toy settings, practical conditional generation tasks such as text-to-image synthesis involve highly complex and multimodal relationships between the condition C and the target data X_1. Modern text-to-image architectures [17, 33, 48] are specifically designed to model such complexity by tightly integrating the conditioning signal into the flow model $v_\theta(\cdot)$, allowing the conditional information to directly modulate the vector field itself. However, this tight integration also makes the optimization of a suitable source distribution more challenging. Since the flow model can account for much of the conditional information directly from C, minimizing the flow matching objective imposes only weak supervision on the source distribution. We hypothesize that this effect is amplified at T2I scale, where encoding compositional text–image information into X_0 is substantially harder than in our toy settings (Fig. 2). Conversely, removing C from $v_\theta(\cdot)$ restores a condition-discriminative source but weakens the conditional model (Fig. 11 and Tab. 4 in Appx. B).

To fully leverage modern conditional architectures while mitigating this optimization challenge, we introduce an explicit source–target alignment objective. Motivated by recent findings that directional information is critical in high-dimensional flow matching [34], we turn this insight into a training objective for the learned source, adopting a negative cosine similarity loss to encourage directional alignment between learned source and target samples. Unlike an MSE objective, which matches both

[4]The support of a random variable X is defined as the smallest closed set $S \subseteq \mathbb{R}^d$ such that $\mathbb{P}(X \in S) = 1$.

Table 1: **Component-wise analysis on ImageNet 256×256.** We analyze the effect of individual components on the captioned ImageNet-1K dataset. All models are trained for 100K iterations with a batch size of 1024 and evaluated using a 50-step Euler ODE sampler without guidance. Baseline models with a fixed Gaussian source are highlighted in gray, **bold** entries denote the default setting for subsequent experiments, and underlined values indicate the best results. In this experiment, RAE (DINOv2) is used as the image autoencoder. († denotes a baseline model with increased parameters to approximately match the parameter count introduced by the source generator.)

Source Distribution	Text Encoder	Flow Backbone	Align Loss	Reg. Loss	FID ↓	CLIP ↑	IS↑	sFID↓	Prec.↑	Recall↑	FDD↓	VQAScore↑
$\mathcal{N}(0, I)$	CLIP	LightningDiT	✗	✗	3.721	0.3283	169.2	6.175	0.7977	0.5718	87.65	0.2130
		MMDiT			3.412	0.3399	186.1	6.507	0.7906	0.5747	78.39	0.4710
		UnifiedNextDiT			3.036	0.3398	187.0	5.859	0.7917	0.5881	69.08	0.4819
		UnifiedNextDiT†			2.925	0.3396	189.1	5.730	0.7898	0.5974	65.29	0.4797
$p_\phi(X_0 \mid C)$	CLIP	UnifiedNextDiT	✗	✗	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
				KL	2.904	0.3405	190.1	5.671	0.7869	0.5913	65.92	0.4854
				VarReg	2.765	0.3404	195.6	5.630	0.7921	0.5958	60.64	0.4747
	CLIP	UnifiedNextDiT	✗	VarReg	2.765	0.3404	195.6	5.630	0.7921	0.5958	60.64	0.4747
			MSE		2.942	0.3410	195.5	6.247	0.7896	0.5902	69.89	0.5006
			CosSim		2.453	0.3420	203.7	5.491	0.7947	0.6029	54.00	0.5105
	CLIP	LightningDiT	CosSim	VarReg	3.041	0.3363	191.9	5.972	0.7961	0.5791	68.64	0.3283
		MMDiT			3.051	0.3415	199.0	6.339	0.7978	0.5745	69.64	0.4994
		UnifiedNextDiT			2.453	0.3420	203.7	5.491	0.7947	0.6029	54.00	0.5105
	CLIP	UnifiedNextDiT	CosSim	VarReg	2.453	0.3420	203.7	5.491	0.7947	0.6029	54.00	0.5105
	Qwen3				2.519	0.3409	200.5	5.465	0.7953	0.6009	54.84	0.5215

direction and magnitude and thereby restricts source mobility (Tab. 1), this provides source-specific guidance without directly matching target magnitudes or shrinking stochastic support:

$$\mathcal{L}_{\text{align}}(\phi) = \mathbb{E}_{C,X_1}\left[1 - \frac{X_0 \cdot X_1}{\|X_0\|\,\|X_1\|}\right]. \tag{10}$$

Combined with the flow matching loss $\mathcal{L}_{\text{FM}}$ and the variance regularization $\mathcal{L}_{\text{VarReg}}$, the final training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{VarReg}}\mathcal{L}_{\text{VarReg}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}, \tag{11}$$

where λ_{VarReg} and λ_{align} are hyperparameters that balance the distribution support and alignment quality of the learned source distribution, respectively.

5 Experiments

In this section, we first validate the design components that make source learning stable and effective in practice (*how*), then demonstrate the underlying mechanism through training dynamics and flow geometry (*why*), and finally identify the representational conditions under which source learning is most effective (*when*). We conclude by scaling CSFM to a 1.3B model on a modern T2I benchmark.

Implementation details. We use RAE (DINOv2) [73] as the default target image representation and adopt a DDT head [64] to provide sufficient model capacity for the high-dimensional feature space. The motivation for the target representation choice, along with a deeper analysis, is discussed in Sec. 5.3. To facilitate fair evaluation at an appropriate scale for practical text-to-image generation tasks, we construct a benchmark dataset based on ImageNet-1K [50]. To study T2I generation in a standardized setting, we employ Qwen3-VL [3] to generate descriptive captions for images (see Fig. 12), rather than simple class-level captions [14]. Model performance is primarily evaluated using FID [23] and CLIP Score [22], together with Inception Score (IS) [51], sFID, Precision–Recall, FDD [59], and VQAScore [37] metrics on the validation set. For ImageNet-1K evaluation, images are generated from validation captions and evaluated on the full 50K-image validation set. Unless otherwise specified, all evaluations are conducted without any guidance; guidance-related results and details are provided in Appx. F. Additional training and evaluation details are provided in Appx. D.

5.1 Validating design components

Tab. 1 summarizes the component-wise analysis, where each paragraph below corresponds to the rows highlighted in the same color. Additional ablations are provided in Appx. O.

Choice of source regularization matters. We examine the role of the regularization loss $\mathcal{L}_{\text{VarReg}}$ in stabilizing source learning. Using CLIP [49] as the text encoder, we evaluate how variance regularization affects model behavior. Without any regularization, the learned source distribution

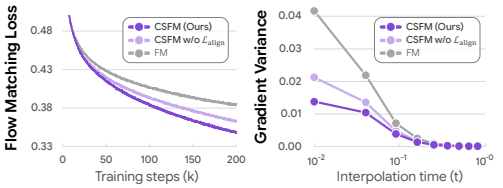

Figure 3: **Flow matching loss and gradient variance ($\mathrm{Var}(\nabla_\theta \mathcal{L}_{\mathbf{FM}})$).** We compare the training dynamics of standard FM, CSFM without alignment loss, and CSFM. CSFM achieves faster loss convergence and lower gradient variance, particularly at early interpolation times near the source. Details of the measurement are provided in Appx. I.

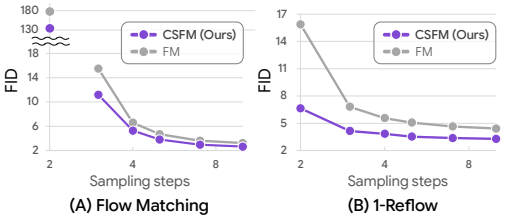

Figure 4: **Few-step generation and flow straightness.** We compare FID across different sampling steps for (A) Flow Matching and (B) 1-Reflow. CSFM degrades more gracefully as the number of steps decreases, indicating reduced path intersections and a straighter transport field compared to the FM baseline.

collapses, driving the training objective near zero and preventing meaningful generation. Standard KL regularization avoids this collapse and improves upon a fixed Gaussian source (FID $3.036 \rightarrow 2.904$), but the gains remain limited. As discussed in Sec. 4.2, this is due to the KL regularization constraining the source mean $\mu_\phi(C)$ toward the origin, which restricts source mobility. In contrast, our variance regularization (FID 2.765) preserves variance control while allowing the mean to adapt freely, enabling effective alignment with the target distribution. This design already yields broad improvements across evaluation metrics, highlighting the importance of unconstrained mean adaptation for learning expressive source distributions in large-scale text-to-image generation.

Alignment is effective in complex settings. We next examine the role of the alignment loss $\mathcal{L}_{\text{align}}$ in improving optimization when learning condition-dependent sources under complex conditioning. As discussed in Sec. 4.3, strong conditioning in the flow model $v_\theta(\cdot)$ can substantially weaken the learning signal available to the source, making optimization unstable and often yielding a poorly trained source distribution (see Appx. B). In this setting, directional alignment substantially improves performance and mitigates these optimization difficulties. Importantly, CSFM without the alignment loss exhibits higher FM loss and increased gradient variance in Fig. 3, indicating that alignment improves flow matching optimization itself rather than acting as an external regularizer. A more detailed discussion of these statistics is provided in Sec. 5.2. To further highlight the importance of directional alignment, we compare this approach with an MSE objective and find that directly minimizing the ℓ_2 distance between X_0 and X_1 overly restricts the source distribution, limiting the flexibility needed for effective source learning under the FM objective.

Robustness to conditioning architecture. We evaluate whether CSFM is consistently effective across different conditioning architectures. We consider three representative paradigms in modern text-to-image generation: LightningDiT [70], which injects conditioning primarily via adaptive layer normalization (AdaLN) [46]; MMDiT [17], which adopts a dual-stream design that processes text and image tokens in separate but interacting branches; and UnifiedNextDiT [48], which employs a unified sequence representation with Multimodal RoPE (mRoPE) [4] to capture cross-modal structure without explicit modulation layers. As shown in Tab. 1, our method consistently improves performance over the baseline across all three architectures. These results indicate that CSFM yields robust gains in text-to-image generation, independent of the specific conditioning mechanism used by the backbone.

Robustness to text encoders. We further assess the robustness of our method by replacing the CLIP text encoder with a large language model (LLM). Specifically, we use Qwen3-0.6B [69] as the text encoder and observe that our method maintains comparable performance to the CLIP-based setting. This result indicates that our framework generalizes across different text encoder architectures and is not tied to a specific encoder design.

Unless otherwise specified, all subsequent experiments adopt the default configuration highlighted in **bold** in Tab. 1.

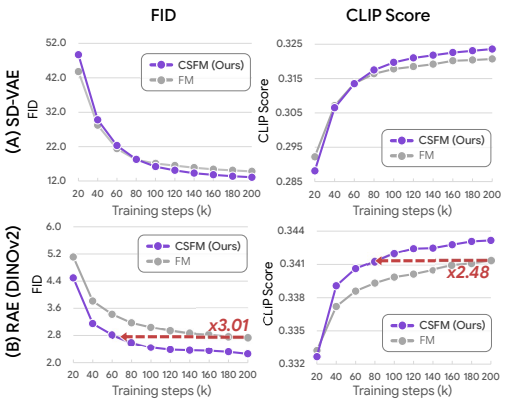

Figure 5: **Training efficiency under different target representations.** We compare FID and CLIP Score trajectories between CSFM and FM, using (A) SD-VAE and (B) RAE (DINOv2) target representations on the ImageNet-1K validation set. While CSFM yields consistent gains under both representations, it substantially accelerates convergence and achieves larger improvements in the structured RAE space.

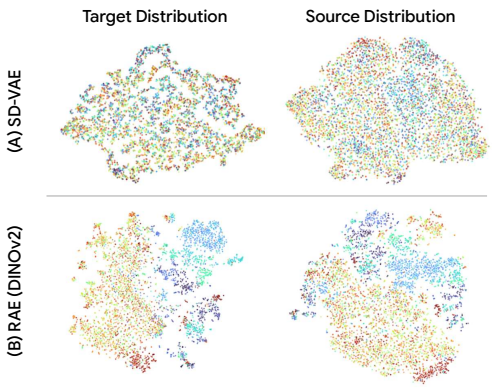

Figure 6: **t-SNE visualization of target and learned source distributions.** We visualize t-SNE embeddings of target (left) and corresponding learned source (right) distributions, colored by class labels, for (A) SD-VAE and (B) RAE (DINOv2) representations. The entangled SD-VAE space leads to an equally entangled source, whereas the structured RAE space enables a semantically organized source distribution.

5.2 Why CSFM improves Flow Matching

Prior work links improved source–target coupling to lower intrinsic variance, fewer path intersections, and straighter flows [35, 47, 60], primarily in unconditional generation. We examine whether these signatures persist when the coupling is induced by a learned condition-dependent source in strongly conditioned T2I models, a regime in which existing condition-dependent source methods [9, 28] have not been analyzed through this lens.

CSFM improves training dynamics of flow matching. By jointly learning a condition-dependent source under the FM loss, CSFM can reduce the intrinsic variance term in Eq. (6), which we directly verify in toy settings in Appx. A.2. This leads to markedly improved optimization dynamics for flow matching. As shown in Fig. 3, CSFM achieves a faster decrease in the FM loss, indicating accelerated convergence [5, 6]. Looking more closely at the optimization behavior, effective source–target coupling is known to reduce the gradient variance of the FM loss [47]. We empirically examine this behavior by measuring the gradient variance at 100K training step in Fig. 3. CSFM consistently attains lower gradient variance than the baseline, with pronounced gains at small interpolation times (i.e., near the source). This further supports that the condition-dependent source provides cleaner velocity supervision and reduces the corresponding ambiguity.

CSFM improves flow matching performance. We show that the improved training dynamics translate into performance gains. As shown in Fig. 5, CSFM achieves consistent improvement, yielding a $3.01\times$ speedup in FID convergence and a $2.48\times$ speedup in CLIP Score in the RAE [73] latent space. We take a deeper look at the performance discrepancy between Stable Diffusion VAE (SD-VAE) and RAE in Sec. 5.3. Qualitatively, CSFM tends to better reflect complex text conditioning involving multiple objects and relationships, while preserving high visual fidelity, as shown in Fig. 16.

CSFM straightens transport paths. Improved source–target coupling can reduce intrinsic variance (Eq. 6) and form straighter transport trajectories [43, 60]. To further investigate the straightness of the learned flow, we also conduct 1-Reflow [41] experiments. We fine-tune their 100K-step checkpoints for 20K steps on sampled pairs $(X_0, X_1^{\text{sampled}})$, freezing the source generator for CSFM.

We first directly assess this geometry using the trajectory straightness error of InstaFlow [42] (see Appx. J and Tab. 9). The straightness error of CSFM is 9.1% and 24.4% lower than that of standard FM before and after Reflow, respectively, indicating that CSFM learns straighter flow trajectories.

In Fig. 4, we show that this improved straightness translates into better few-step generation. When reducing sampling from 50 to 3 steps, FID degrades by 8.75 for CSFM compared with 12.47 for

Table 2: **Comparison between condition-dependent source and coupling methods on ImageNet 256×256.** We compare methods that learn a text-dependent source distribution or perform text-aware coupling. All baselines are reproduced adopting RAE (DINOv2) for target latent using identical model architectures for fair comparison and evaluated with 50-step Euler ODE sampler.

Method	FID ↓	CLIP ↑	IS ↑	FDD ↓	VQAScore ↑
Standard FM	3.036	0.3398	187.0	69.08	0.4819
CrossFlow [40]	2.957	0.3301	182.8	66.21	0.2613
C^2OT [13]	3.365	0.3406	184.3	78.72	0.4890
CPD [28]	3.219	0.3327	190.2	80.48	0.4113
CAR-Flow [9]	2.929	0.3408	192.8	66.38	0.4872
CSFM (Ours)	**2.453**	**0.3420**	**203.7**	**54.00**	**0.5105**

Table 3: **Large-scale text-to-image evaluation.** Results on GenEval and DPG-Bench. We report standard FM and CSFM with UnifiedNextDiT (1.3B), together with results from prior work that evaluate the same model families at different parameter scales, to contextualize the magnitude of performance gains at this scale.

Model	#Params	GenEval	DPG-Bench
BLIP3o [11]	4B	0.81	79.36
	8B	0.84	81.60
Sana [67]	0.6B	0.64	83.60
	1.6B	0.66	84.80
Standard FM	1.3B	0.77	78.31
CSFM (Ours)	1.3B	**0.80**	**81.11**

standard FM (Fig. 4 (A)). After 1-Reflow, reducing sampling from 50 to 2 steps degrades FID by only 3.51 for CSFM versus 11.75 for standard FM (Fig. 4 (B)). These results demonstrate that CSFM learns straighter trajectories, with practical benefits for few-step generation.

5.3 When source learning helps: Target representation matters

In Fig. 5, CSFM improves performance across image representation spaces, but with much larger gains in RAE (DINOv2) [73] latents than in SD-VAE [15]. This indicates that the structure of the target representation is critical for effective source learning. We further verify that this trend generalizes to another source-learning method in Appx. N.

At a high level, learning a condition-dependent source is most beneficial when the conditioning signal C induces a well-separated and discriminative structure in the target space, as illustrated in Fig. 2 and Fig. 9. In such cases, samples sharing the same condition form relatively compact clusters, making the source mean $\mu_\phi(C)$ well-defined and easier to align with the target distribution. This reduces path intersections, yielding more coherent flow-matching supervision.

However, this advantage diminishes when a single condition corresponds to a highly multimodal target distribution. As shown in Appx. A.4, when samples sharing the same C are spread across distant modes, the source mean becomes ambiguous. This ambiguity leads to frequent path intersections, conflicting supervision, and persistently high intrinsic variance. In such cases, the learned source behaves more like a fixed Gaussian prior and provides limited additional benefit.

This property makes CSFM most effective with representation autoencoders operating in structured latent spaces, such as DINOv2 [45] or SigLIP2 [63]. In these spaces, samples sharing the same condition tend to be more concentrated [7, 27, 66], reducing multimodality with respect to C. This concentration simplifies source learning and improves the effectiveness of CSFM.

We further support this analysis with t-SNE visualizations in Fig. 6, where points are color-coded by their corresponding classes. In the SD-VAE representation, the target distribution exhibits strong entanglement and weak structure with respect to the conditioning signal, and this lack of structure is reflected in a similarly non-discriminative learned source (Fig. 6 (A)). In contrast, the RAE (DINOv2) representation induces a more organized target geometry, which in turn allows the source distribution to become more discriminative and results in larger performance gains (Fig. 6 (B)).

5.4 Comparison with condition-dependent source and coupling methods

We compare CSFM with C^2OT [13], CPD [28], CrossFlow [40], and CAR-Flow [9] in Tab. 2, using the same RAE (DINOv2) target representation and UnifiedNextDiT architecture for fair comparison. Although these methods modify either the coupling or the source distribution, they provide limited gains in this high-dimensional T2I setting and often fail to improve over the Gaussian-source baseline. In contrast, CSFM achieves the best performance across fidelity and alignment metrics. This suggests that, at modern T2I scale, the benefit of source conditioning depends critically on how the source is trained and regularized under the FM objective. Details of this comparison are provided in Appx. E.

5.5 Scaling CSFM to modern text-to-image scale

To examine whether CSFM remains effective at scale, we scale the default configuration in Tab. 1 to a 1.3B-parameter model and replace the text encoder with Qwen3-0.6B for longer text inputs. The model is pretrained on the BLIP3o pretraining dataset [11], which contains approximately 36M samples, and then fine-tuned on BLIP3o-60K. We use a SigLIP2-based RAE decoder [62] at 224×224 resolution and evaluate on GenEval [18] and DPG-Bench [26]. As shown in Tab. 3, CSFM consistently outperforms the Gaussian-source baseline across benchmarks, demonstrating that learnable source distributions remain effective for high-capacity text-to-image generation. Full benchmark results are provided in Appx. P.

However, as standard text-to-image benchmarks become increasingly saturated at this scale, quantitative metrics provide only a limited view of model behavior. We therefore present qualitative comparisons in Fig. 1, Fig. 17, and Fig. 18, illustrating perceptual differences induced by source design and providing complementary evidence for the benefits of our approach in large-scale settings.

6 Conclusion

In this work, we present *Condition-dependent Source Flow Matching* (CSFM), treating the source distribution as a learnable design choice for conditional flow matching. We show *why* source learning helps through the lens of intrinsic-variance-related ambiguity, *how* stable end-to-end training can be achieved with variance-only regularization and directional alignment, and *when* the gains are most pronounced in semantically structured target spaces. Extensive experiments demonstrate faster convergence, straighter flows, and consistent improvements at modern text-to-image scale.

Acknowledgments and Disclosure of Funding

We are grateful to RAE [73] for the excellent codebase, which served as the launching point for our research. We also thank Saining Xie and Sihyun Yu for reviewing this manuscript and providing constructive feedback. This research was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, RS-2024-00509279, RS-2025-II212068, RS2023-00227592, RS-2025-02214479, RS-2024-00457882, RS-2025-25441838, RS2025-02217259, RS-2026-25519202, RS-2024-00436680) and the Culture, Sports, and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (RS-2024-00345025, RS-2024-00333068), the InnoCORE program of the Ministry of Science and ICT (N10250156) and National Research Foundation of Korea (RS-2024-00346597). This work was also supported in part through the NYU IT High Performance Computing (HPC) resources, services, and staff expertise.

References

[1] Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A noise is worth diffusion guidance, 2024. URL `https://arxiv.org/abs/2412.03895`.

[2] Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. *arXiv preprint arXiv:2408.14608*, 2024.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023.

[6] Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *arXiv preprint arXiv:2305.16860*, 2023.

[7] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL `https://arxiv.org/abs/2104.14294`.

[9] Chen Chen, Pengsheng Guo, Liangchen Song, Jiasen Lu, Rui Qian, Xinze Wang, Tsu-Jui Fu, Wei Liu, Yinfei Yang, and Alex Schwing. Car-flow: Condition-aware reparameterization aligns source and target for better flow matching. *arXiv preprint arXiv:2509.19300*, 2025.

[10] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

[11] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[12] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[13] Ho Kei Cheng and Alexander Schwing. The curse of conditions: Analyzing and improving optimal transport for conditional flow-based generation. *arXiv preprint arXiv:2503.10636*, 2025.

[14] Lucas Degeorge, Arijit Ghosh, Nicolas Dufour, David Picard, and Vicky Kalogeiton. How far can we go with imagenet for text-to-image generation? *arXiv preprint arXiv:2502.21318*, 2025.

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[16] Zibin Dong, Yicheng Liu, Yinchuan Li, Hang Zhao, and Jianye Hao. Conditioning matters: Training diffusion policies is faster than you think. *Advances in Neural Information Processing Systems*, 38: 119651–119676, 2026.

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

[18] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL `https://arxiv.org/abs/2006.07733`.

[20] Doron Haviv, Aram-Alexandre Pooladian, Dana Pe'er, and Brandon Amos. Wasserstein flow matching: Generative modeling over families of distributions. *arXiv preprint arXiv:2411.00698*, 2024.

[21] Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. Flowtok: Flowing seamlessly across text and image tokens. *arXiv preprint arXiv:2503.10772*, 2025.

[22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

[23] 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. *Advances in neural information processing systems*, 30, 2017.

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

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[27] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[28] Noam Issachar, Mohammad Salama, Raanan Fattal, and Sagie Benaim. Designing a conditional prior distribution for flow-based generative models. *arXiv preprint arXiv:2502.09611*, 2025.

[29] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

[30] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL `https://arxiv.org/abs/1312.6114`.

[32] Lingkai Kong, Molei Tao, Yang Liu, Bryan Wang, Jinmiao Fu, Chien-Chih Wang, and Huidong Liu. Alignflow: Improving flow-based generative models with semi-discrete optimal transport, 2025. URL `https://arxiv.org/abs/2510.15038`.

[33] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[34] Junho Lee, Kwanseok Kim, and Joonseok Lee. Is there a better source distribution than gaussian? exploring source distributions for image flow matching. *arXiv preprint arXiv:2512.18184*, 2025.

[35] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In *International Conference on Machine Learning*, 2023.

[36] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[37] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024.

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

[39] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

[40] Qihao Liu, Xi Yin, Alan Yuille, Andrew Brown, and Mannat Singh. Flowing from words to pixels: A noise-free framework for cross-modality evolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2755–2765, 2025.

[41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[42] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The twelfth international conference on learning representations*, 2024.

[43] Chenrui Ma, Xi Xiao, Tianyang Wang, Xiao Wang, and Yanning Shen. Learning straight flows: Variational flow matching for efficient generation, 2025. URL https://arxiv.org/abs/2511.17583.

[44] Chaojie Mao, Chen-Wei Xie, Chongyang Zhong, Haoyou Deng, Jiaxing Zhao, Jie Xiao, Jinbo Xing, Jingfeng Zhang, Jingren Zhou, Jingyi Zhang, et al. Wan-image: Pushing the boundaries of generative visual intelligence. *arXiv preprint arXiv:2604.19858*, 2026.

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[46] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[47] 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, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28100–28127. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/pooladian23a.html.

[48] Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Jiakang Yuan, Xinyue Li, Dongyang Liu, et al. Lumina-image 2.0: A unified and efficient image generative framework. *arXiv preprint arXiv:2503.21758*, 2025.

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

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

[51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[52] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[53] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qy07OHsJT5.

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

[55] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

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

[57] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[58] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[59] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

[60] 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. *arXiv preprint arXiv:2302.00482*, 2023.

[61] Alexander Tong, Nikolay Malkin, Kilian FATRAS, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. URL `https://openreview.net/forum?id=adkj23mvB0`.

[62] Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

[63] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[64] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025.

[65] Xihua Wang, Xin Cheng, Yuyue Wang, Ruihua Song, and Yunfeng Wang. Vaflow: Video-to-audio generation with cross-modality flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11777–11786, October 2025.

[66] Evžen Wybitul, Javier Rando, Florian Tramèr, and Stanislav Fort. Representations of text and images align from layer one. *arXiv preprint arXiv:2601.08017*, 2026.

[67] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[68] Sihan Xu, Ziqiao Ma, Wenhao Chai, Xuweiyi Chen, Weiyang Jin, Joyce Chai, Saining Xie, and Stella X. Yu. Next-embedding prediction makes strong vision learners, 2025. URL `https://arxiv.org/abs/2512.16922`.

[69] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[70] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[71] 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. *arXiv preprint arXiv:2410.06940*, 2024.

[72] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14730–14740, 2023.

[73] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[74] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

Appendix

A Toy experiments

A.1 Toy experiments setup

We conduct toy experiments (Fig. 2) on two standard two-dimensional synthetic datasets, Eight Gaussians with polar angle condition and Two Moons with x-coordinate condition, using the official C^{2}OT implementation [13]. For the conditional flow model, the condition, time and the coordinate x_t are projected into a shared hidden dimension, summed and passed through a 64-dimensional, 10-layer MLP with GELU activations. For the source generator, we use a 3-layer MLP, also with GELU activations. We train the network for 20K optimization steps with a batch size of 256, using Adam [30] with a learning rate of 3×10^{-4}. For visualization in Fig. 2, we generate samples using an Euler sampler with 16 integration steps.

A.2 Intrinsic variance comparison

We estimate the intrinsic variance $\mathbb{E}[\mathrm{Var}(\Delta \mid X_t)]$ during training for both the fixed Gaussian source flow model and the learnable source flow model, as shown in Fig. 7. Under a fixed Gaussian source, this term is determined by the prescribed coupling and remains essentially unchanged in expectation during training. In contrast, jointly training the source and flow model makes the coupling adaptive under the flow-matching objective, reducing the intrinsic variance over training.

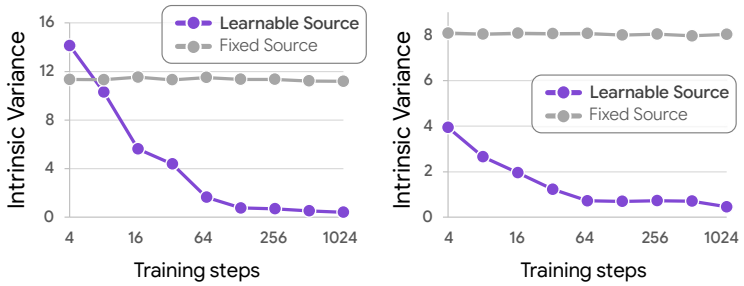

Figure 7: **Estimated intrinsic variance** for fixed Gaussian source (Fig. 2 (A)) and learnable source (Fig. 2 (E)).

Intrinsic variance estimation. We directly estimate the intrinsic variance term $\mathbb{E}[\mathrm{Var}(\Delta \mid X_t)]$, where $\Delta = X_1 - X_0$ and $X_t = (1-t)X_0 + tX_1$, in the toy experiments. At each evaluation checkpoint, we construct a pool of $N = 100{,}000$ source–target pairs using the same coupling as in training, sample $t \sim \mathrm{Uniform}(0, 1)$, and compute the corresponding (X_t, Δ). Since exact conditioning on the same X_t is not possible in continuous space, we approximate the local conditional distribution using nearest neighbors in X_t-space. Specifically, for $M = 10{,}000$ query points, we define $\mathcal{N}_K(i)$ as the set of the $K = 64$ nearest neighbors of the i-th query point in the pool, measured in X_t-space and excluding the query point itself, and compute

$$\widehat{\mathrm{IV}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{K} \sum_{j \in \mathcal{N}_K(i)} \left\| \Delta_j - \bar{\Delta}_i \right\|_2^2, \qquad \bar{\Delta}_i = \frac{1}{K} \sum_{j \in \mathcal{N}_K(i)} \Delta_j.$$

This estimates the average local variance of the velocity targets around each intermediate point X_t.

A.3 Condition-dependent source with an unconditional flow model

We conduct additional toy experiments using an *unconditional* flow model to investigate how the learnable condition-dependent source distribution is influenced by conditioning injected into the flow model. Specifically, we compare a conditional source trained with an unconditional flow model

$v_\theta(X_t, t)$ against one trained with a conditional flow model $v_\theta(X_t, t, C)$. Note that alignment loss in Sec. 4.2 is not applied for toy experiments.

As shown in Fig. 8 (B), when the flow model does not receive the condition C, the learned source distribution becomes noticeably more discriminative across conditions. This is mainly because in this setting, the source generator must encode condition-specific structure in order to reduce the flow matching loss, resulting in a more informative and well-separated conditional source. However, with unconditional flow model, the vector field is shared across conditions, making it harder for transport trajectories to branch at similar spatial locations.

This experiment should be viewed as a diagnostic for the role of the learned source. In the unconditional-flow variant, the condition can influence the velocity target only through the sampled source $X_0 \sim p_\phi(X_0 \mid C)$ and the resulting intermediate state $X_t = (1-t)X_0 + tX_1$, making the reduction of source-induced ambiguity directly visible in the learned source. In the main conditional model, $v_\theta(X_t, t, C)$ also receives C, so part of the conditional structure can be absorbed by the velocity network, weakening the learning signal to the source. However, with finite-capacity networks and continuous high-dimensional conditions, making X_t itself condition-compatible can still ease the supervision seen by the flow model. This motivates the explicit alignment objective in Sec. 4.3, which encourages the source to retain useful condition-dependent structure during joint training. We further illustrate this optimization effect in a practical setting in Appx. B.

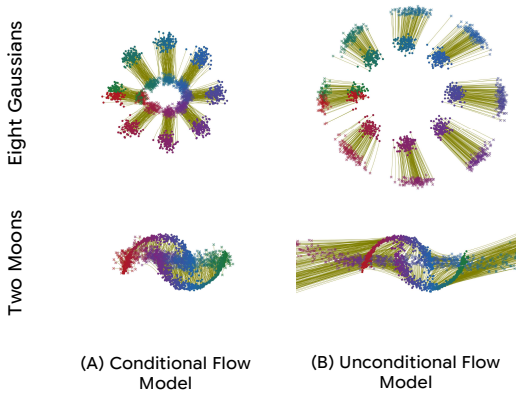

Figure 8: **Learned conditional source distributions with conditional vs. unconditional flow models.**

A.4 Ill-conditioned cases

We further investigate ill-conditioned scenarios in Fig. 9, where the conditioning signal does not sufficiently concentrate the target distribution. Specifically, we consider two extreme cases in which the condition is defined by the ℓ_2 norm of the modes or by the x-coordinate in the Eight Gaussians setting. In these cases, a single condition value can correspond to multiple spatial locations, making the placement of a suitable source inherently ambiguous. Consequently, the learned source gravitates toward an averaged location—for instance, the midpoint between modes under the x-coordinate condition, or the center of the distribution in the extreme ℓ_2-norm case—resulting in a more Gaussian-like source.

Despite still encouraging straighter transport paths compared to a fixed Gaussian prior, the learned source distribution in this setting becomes less discriminative across conditions, effectively reverting toward a Gaussian-like baseline (Fig. 9 (A)). This observation aligns with the representation analysis in Sec. 5.3: learnable, condition-dependent source distributions are most effective when the target representation provides sufficient structure to reduce multimodality with respect to the condition.

A.5 Variance explosion cases: stopping gradient on target velocity

In the context of self-supervised learning [8, 19, 68], stopping gradients is a commonly adopted strategy to prevent representational collapse. Motivated by this practice, one may consider stopping the gradient on the target velocity in Eq. 7 when jointly learning the flow model and the source

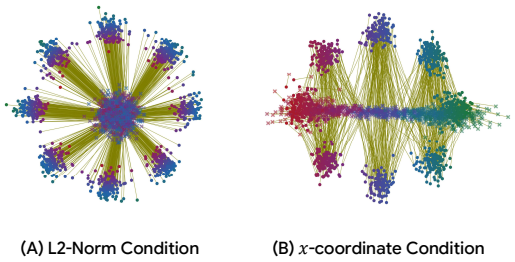

Figure 9: **Learned conditional source distributions under ill-conditioned settings.**

generator. Concretely, this corresponds to treating the target velocity $\Delta = \mathrm{sg}(X_1 - X_0)$ as a constant with respect to the source parameters, where sg denotes the stop-gradient operation.

However, as shown in Fig. 10, we find that stopping gradients can cause the explosion of the variance of the learnable source distribution in some cases. This behavior corresponds to a degenerate solution in which the source generator trivially reduces the path variance by allowing the source samples to diverge.

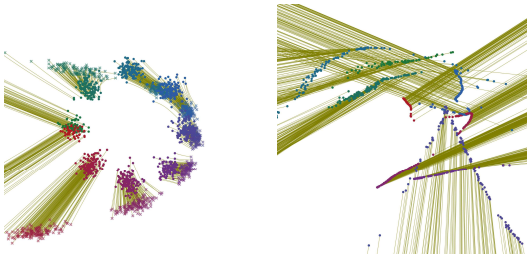

Figure 10: **Variance explosion case.** The variance of the learned source distribution diverges under an unconditional flow model for (B) $\lambda_{\mathrm{VarReg}} = 1.0$.

B Direct source–target alignment for complex training dynamics

As discussed in Sec. 4.2, conditioning the flow model $v_\theta(\cdot)$ with condition C (i.e., $v_\theta(X_t, t, C)$) can make source learning more challenging from an optimization perspective in complex text-to-image settings. As observed in Appx. A.3, we attribute this to the fact that the source model has less incentive to remain discriminative when conditional information is directly handled by the flow model.

This effect also manifests in practical settings. We evaluate this behavior using LightningDiT with a learnable source generator and the variance regularization term $\mathcal{L}_{\mathrm{VarReg}}$. As shown in Tab. 4, adding conditioning to the flow model $v_\theta(\cdot)$ degrades FID relative to the unconditioned counterpart, despite improving CLIP score due to stronger conditional modeling. Consistently, Fig. 11 illustrates that conditioning $v_\theta(\cdot)$ causes the learned source distribution to become less discriminative, indicating that the source struggles to learn meaningful structure.

In contrast, when combined with the alignment loss, we observe improvements in both FID and CLIP score, along with a more discriminative and structured source distribution.

Table 4: **Ablation results analyzing the effects of alignment loss and backbone conditioning.**

Flow backbone	Condition C on v_θ	Alignment Loss $\mathcal{L}_{\mathrm{align}}$	FID	CLIP
LightningDiT	✗	✗	3.167	0.3335
	✓	✗	3.188	0.3354
	✓	✓	3.041	0.3363

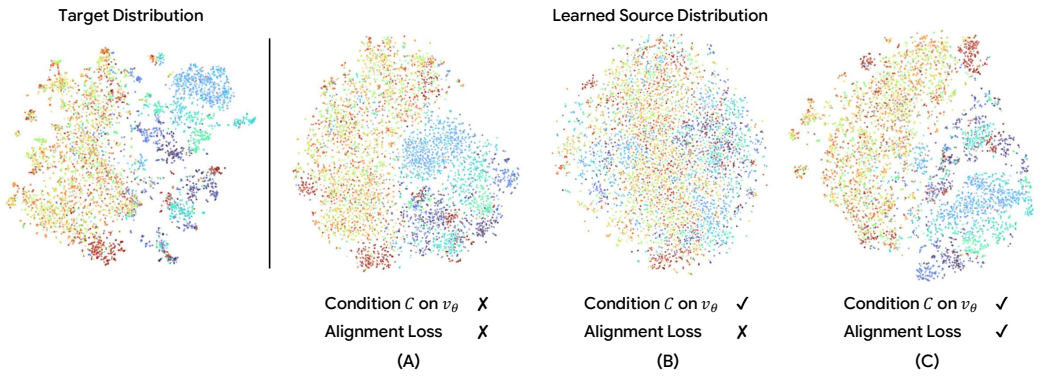

Figure 11: **t-SNE visualization** of learned source distribution comparing the effects of condition injection and alignment loss.

C Decomposition of the Flow Matching Objective

We derive the decomposition of the Flow Matching (FM) objective in Eq. (4) into an approximation error term and an intrinsic variance term. Recall that the FM loss is defined as:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim p(t),\,(X_0,X_1)\sim\pi}\big[\|v_\theta(X_t,t) - \Delta\|^2\big], \tag{12}$$

where $\Delta = X_1 - X_0$ and $X_t = (1-t)X_0 + tX_1$. Let

$$u_t(x) := \mathbb{E}[\Delta \mid X_t = x] \tag{13}$$

denote the marginal velocity field induced by the coupling π. We rewrite the squared error as:

$$\|v_\theta(X_t,t) - \Delta\|^2 = \|v_\theta(X_t,t) - u_t(X_t) + u_t(X_t) - \Delta\|^2. \tag{14}$$

Expanding the norm yields:

$$\begin{aligned}\|v_\theta(X_t,t) - \Delta\|^2 = \|v_\theta(X_t,t) - u_t(X_t)\|^2 + \|u_t(X_t) - \Delta\|^2 \\ + 2\langle v_\theta(X_t,t) - u_t(X_t),\, u_t(X_t) - \Delta\rangle.\end{aligned} \tag{15}$$

Taking expectation with respect to t and $(X_0, X_1) \sim \pi$, the cross term vanishes:

$$\begin{aligned}&\mathbb{E}\big[\langle v_\theta(X_t,t) - u_t(X_t),\, u_t(X_t) - \Delta\rangle\big] \\ &\quad= \mathbb{E}\Big[\mathbb{E}\big[\langle v_\theta(X_t,t) - u_t(X_t),\, u_t(X_t) - \Delta\rangle \mid X_t\big]\Big] \\ &\quad= \mathbb{E}\Big[\langle v_\theta(X_t,t) - u_t(X_t),\, \mathbb{E}[u_t(X_t) - \Delta \mid X_t]\rangle\Big] = 0,\end{aligned} \tag{16}$$

since $\mathbb{E}[\Delta \mid X_t] = u_t(X_t)$ by definition. The remaining second term satisfies

$$\mathbb{E}\big[\|u_t(X_t) - \Delta\|^2\big] = \mathbb{E}\big[\mathrm{Var}(\Delta \mid X_t)\big], \tag{17}$$

which follows directly from the definition of conditional variance. Combining the above results, the FM objective decomposes as:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,\pi}\big[\|v_\theta(X_t,t) - u_t(X_t)\|^2\big] + \mathbb{E}_{t,\pi}\big[\mathrm{Var}(\Delta \mid X_t)\big]. \tag{18}$$

The first term corresponds to the approximation error incurred by learning the marginal velocity field u_t, while the second term is an intrinsic variance determined solely by the coupling π, independent of the model parameters θ.

For conditional FM, the same orthogonal decomposition holds after additionally conditioning on C. Defining $u_t(x,c) = \mathbb{E}[\Delta \mid X_t = x, C = c]$, the intrinsic term becomes

$$\mathbb{E}_{t,\pi(X_0,X_1,C)}[\mathrm{Var}(\Delta \mid X_t, C)].$$

This term captures the ambiguity that remains when the condition is treated as fully available to the velocity model. In the main text, we present the standard form in Eq. (6) because it highlights how the interpolation path changes the information and ambiguity carried by X_t itself.

D Training and evaluation details

We detail the architectures of the source generator in Tab. 5. The architectures of the flow matching backbones used in Tab. 1 and Tab. 3 are provided in Tab. 6. For UnifiedNextDiT(1.3B), which is employed in the scaling experiments, we omit the DDT head since its hidden dimensionality is sufficiently large relative to the latent dimension. In the scaling experiments, the model is pretrained for 400K iterations and subsequently finetuned for 15K iterations. Note that due to differences in internal architectural design, models with the same number of layers and hidden dimensionality may still have different total parameter counts.

All models are trained on a TPUv5p cluster with Torch/XLA, with each model taking approximately one day to train. For evaluation, all experiments are conducted on NVIDIA L40S GPUs, and images are generated from validation captions for all reported metrics. FID, Inception Score (IS), sFID, and Precision–Recall metrics are computed using ADM precomputed statistics [15]. CLIP Score [22] is computed using the ViT-B/32 model. FDD [59] is computed using DINOv2 features to provide an additional fidelity-oriented evaluation. VQAScore [37] is computed with Qwen2.5-VL to further assess text–image alignment. To balance the scale of source-related objectives with the flow matching loss, we set $\lambda_{\mathrm{VarReg}} = 5.0$ and $\lambda_{\mathrm{align}} = 1.0$ across all experiments; a sensitivity analysis is provided in Appx. O.

For the source generator, we take the input text embedding $\mathbf{e}^{\mathrm{text}} \in \mathbb{R}^{N\times d}$, where N denotes the text sequence length and d is the text embedding dimension. We adopt a Perceiver-style architecture [36] with S learnable query tokens, where S matches the flattened image-token sequence length. The image embedding $\mathbf{e}^{\mathrm{img}} \in \mathbb{R}^{S\times D}$ serves as the target representation, where D is the image embedding dimension. Since flow matching requires the source and target to lie in the same space, we use cross-attention to map the query tokens conditioned on $\mathbf{e}^{\mathrm{text}}$ into the image embedding space $\mathbb{R}^{S\times D}$.

Following a VAE-style parameterization [31], the source generator uses separate output heads to predict the conditional mean $\mu_\phi(C) \in \mathbb{R}^{S\times D}$ and variance $\sigma^2_\phi(C) \in \mathbb{R}^{S\times D}$. We sample $X_0 \sim \mathcal{N}(\mu_\phi(C), \sigma^2_\phi(C)I)$ using the reparameterization trick. For simplicity and to avoid introducing an additional scale hyperparameter, the variance regularization in Eq. (9) uses unit target variance, i.e., target standard deviation 1.0.

For architectures such as LightningDiT [70] and MMDiT [17] that require AdaLN-based modulation, we use the pooled text representation when available; otherwise, we fall back to a mean-pooled text embedding.

At inference, the source generator is evaluated only once to sample the initial source X_0. For 256×256 image generation with 50 NFE, the default sampling setting used throughout the paper, this adds only 0.36 ms of average latency, which is negligible compared with the 91.83 ms average ODE sampling time of the velocity model.

Table 5: **Architectural details of the source generator used in our experiments.**

	Source Generator	
Architecture		
Input dim.	768	1024
Hidden dim.	768	768
Num. layers	8	8
Num. heads	12	12
Params	77M	78M
Text Encoder	CLIP	Qwen3
Max length	77	128

E Comparison implementation and interpretation

We provide implementation details for the comparison methods reported in Tab. 2. For each method, we follow the official implementation as closely as possible while adapting it to the shared RAE

Table 6: **Architectural and optimization details of flow matching backbones used in our experiments.**

	LightningDiT	MMDiT	UnifiedNextDiT	UnifiedNextDiT†	UnifiedNextDiT(1.3B)
Architecture					
Downsample Ratio	16×16	16×16	16×16	16×16	14×14
Latent dim.	768	768	768	768	1152
Num. layers	12	12	14	16	20
Hidden dim.	768	768	864	864	1728
Num. heads	12	12	12	12	16
DDT Head					
Num. layers	2	2	2	2	
Hidden dim.	2048	2048	2048	2048	✗
Num. heads	16	16	16	16	
Params	294M	418M	414M	471M	1.24B
Base Encoder	DINOv2	DINOv2	DINOv2	DINOv2	SigLIP2
Optimization					
Batch Size	1024	1024	1024	1024	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Base lr	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
(β_1, β_2)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)

(DINOv2) target representation and UnifiedNextDiT [48] backbone with a DDT head [64]. All models are trained for 100K steps with batch size 1024 and learning rate 2×10^{-4}, using CLIP [49] as the text encoder. For C^2OT [13], we use an oversampling ratio of 10 for the OT coupling batch. For CPD [28], we pretrain the source generator for 100K steps before training the flow model. For CrossFlow, CPD, and CAR-Flow, we match the number of parameters used in the condition-dependent source generator.

The results in Tab. 2 suggest that simply introducing condition information into the source is not sufficient for robust gains in this high-dimensional T2I setting. CPD constructs the source through a separate regression objective before FM training; while this improves source–target proximity, the source is not optimized with the velocity model under the FM objective. CAR-Flow jointly optimizes a condition-aware reparameterization, but its constrained source parameterization provides limited flexibility in adapting the stochastic support of the source. C^2OT directly modifies sample-level coupling, but reliable minibatch matching becomes difficult at this scale. CrossFlow demonstrates that text-dependent sources are feasible, but its cross-modal transport formulation does not fully exploit the strongly conditioned T2I backbone used here. In contrast, CSFM learns the source under the FM objective while explicitly stabilizing source variance and strengthening source supervision, leading to the strongest performance across fidelity and alignment metrics.

F Guidance

Since CSFM retains a conditional flow backbone $v_\theta(X_t, t, C)$, classifier-free guidance (CFG) [24] can be applied at the backbone level. However, such backbone-level CFG removes the condition only from $v_\theta(\cdot)$, while the source $X_0 \sim p_\phi(X_0 \mid C)$ remains condition-dependent, and thus differs from conventional CFG. A fully unconditional branch could be obtained by additionally dropping the condition from the source through dropped- or mismatched-condition training, as in CrossFlow [40], but we leave this extension unexplored since it is orthogonal to our focus on condition-dependent source learning. We therefore focus on AutoGuidance [29], which guides with a weaker model without requiring an unconditional branch and has been observed to be effective in RAE feature spaces [73].

We select guidance scales by grid search under the same evaluation protocol. As shown in Tab. 7, AutoGuidance improves both the Gaussian-source FM baseline and CSFM, indicating that guidance remains effective with learned condition-dependent sources. For reference, we also report a CFG-guided Gaussian-source FM baseline: CFG improves unguided FM, but remains worse than unguided CSFM in FID. This relatively modest CFG gain is consistent with prior observations in RAE latent spaces [54, 73]. Its higher CLIP Score should be interpreted carefully, since it exceeds the validation-

caption CLIP Score of 0.3452 and may reflect over-optimization toward the CLIP objective rather than uniformly better generation quality.

Table 7: **Guidance analysis on ImageNet 256×256.** We evaluate whether guidance remains effective with learned condition-dependent sources by applying AutoGuidance (AG) [29] to both the Gaussian-source FM baseline and CSFM. AG improves both methods, while CSFM with AG achieves the best FID and maintains a strong CLIP Score. For reference, we also report CFG on the Gaussian-source baseline.

Method	FID ↓	CLIP ↑
Standard FM	3.036	0.3398
Standard FM + CFG	2.801	**0.3463**
Standard FM + AG	2.673	0.3400
CSFM (Ours)	2.453	0.3420
CSFM (Ours) + AG	**2.016**	0.3433

G Transferability of the Learned Source

We investigate whether a source learned with a smaller proxy can be reused by a larger flow model. We transfer only the source-generator parameters from a 7-layer proxy to a newly initialized 14-layer backbone, either frozen or jointly trainable. All settings use captioned ImageNet-1K with the same text encoder and target representation, batch size 512, and 40K iterations per stage. Transfer includes an additional proxy pretraining stage, so its total compute differs from training from scratch.

Table 8: **Source transfer across flow-model capacities.** Results on captioned ImageNet-1K. (P) denotes the 7-layer proxy CSFM.

Method	Flow model depth	Source	CLIP↑	FID↓
Standard FM	14	Fixed Gaussian	0.3346	3.855
CSFM	14	Learnable	0.3361	3.403
(P) Proxy CSFM	7	Learnable	0.3318	4.081
(P → F) Frozen transfer	14	Frozen (from P)	0.3367	3.306
(P → L) Learnable transfer	14	Learnable (from P)	**0.3370**	**3.232**

Tab. 8 shows that frozen transfer improves FID over CSFM from scratch ($3.403 \rightarrow 3.306$), and joint updates further improve it to 3.232. This demonstrates that the learned source can be reused without further optimization.

One hypothesis is that the smaller proxy encourages the source to encode more condition-dependent structure (Sec. 4.3 and Appx. B). However, two-stage optimization and the short training horizon may also explain the gains; their persistence at longer horizons remains unverified. Cross-dataset and cross-representation transfer are left for future work.

H ImageNet captioned dataset

As modern text-to-image frameworks increasingly emphasize scaling, there is no well-established setting for component-wise analysis [14]. To enable controlled component-wise experiments at a manageable scale, we construct a captioned dataset based on ImageNet-1K [50]. Detailed image captions are generated using Qwen3-VL-8B-Instruct [3]. Examples from the resulting dataset are shown in Fig. 12. The captions are publicly released at `https://huggingface.co/datasets/junwann/CSFM-ImageNet1K-Caption`.

I Gradient variance

While we can directly estimate intrinsic variance in the low-dimensional toy setting (Appx. A), doing so in high-dimensional continuous latent spaces is difficult, since it would require reliable samples



Figure 12: **Examples of detailed image–caption pairs from the constructed captioned ImageNet-1K dataset.**

with the same, or sufficiently nearby, intermediate state X_t. Such estimates are highly sensitive to neighborhood choice and become unreliable as dimensionality increases. We therefore use gradient variance as a practical diagnostic of supervision ambiguity induced by the source–target coupling.

We report the variance of gradients of the flow model $v_\theta(\cdot)$ induced by the Flow Matching loss, i.e., $\mathrm{Var}(\nabla_\theta \mathcal{L}_{\mathrm{FM}})$, across the interpolation time t (Fig. 3). To compute gradients, we select five attention layers uniformly sampled from the flow backbone and measure the gradients of the MLP parameters that project inputs to the query, key, and value vectors. The reported variance is further averaged across the selected layers. Due to the small magnitude of individual gradient components, we compute the variance element-wise and report the sum across dimensions. Experiments are conducted using 20K samples from the ImageNet-1K [50] training set and a model trained for 100K steps. We use `torch.func.vmap` to efficiently compute per-sample gradients.

J Trajectory straightness comparison

Following InstaFlow [42], we directly measure trajectory straightness using

$$\mathcal{S}(Z) = \frac{1}{d}\int_0^1 \mathbb{E}_{C,Z_0}\left[\|(Z_1 - Z_0) - v_\theta(Z_t, t, C)\|_2^2\right] \mathrm{d}t, \tag{19}$$

where Z_t follows the generated ODE trajectory $\mathrm{d}Z_t/\mathrm{d}t = v_\theta(Z_t, t, C)$, Z_1 is its generated endpoint, and d is the latent dimensionality. The expectation is taken over evaluation conditions and initial samples from each method's source distribution. We use 50-step Euler integration and 50K samples for both methods. The metric measures the deviation of the instantaneous velocity from the constant endpoint displacement and equals zero for a perfectly straight, constant-velocity trajectory. As shown in Tab. 9, CSFM reduces the straightness error by 9.1% and 24.4% before and after 1-Reflow, respectively.

Table 9: **Trajectory straightness (↓).** We evaluate both methods using 50-step Euler integration and 50K samples per setting.

Method	Flow Matching	1-Reflow
Standard FM	0.1498	0.0270
CSFM (Ours)	**0.1361**	**0.0204**

K Detailed related work

Flow Matching. Flow Matching (FM) learns a velocity field that defines a continuous probability path between a source distribution and a target distribution [38, 39, 41]. Owing to its conceptual simplicity and strong generative performance, FM has become a widely used framework for generative

modeling. Unlike conventional diffusion formulations [25], FM does not restrict the source distribution to a standard Gaussian [40]. Despite this flexibility, most practical systems still adopt a fixed Gaussian source, leaving the design and learning of source distributions comparatively underexplored.

Optimal Transport coupling. A standard independent coupling between Gaussian source samples and target data can induce intersecting training trajectories, leading to curved flows and high-variance FM gradients. To mitigate this issue, OT-based methods couple source and target samples using minibatch-level optimal transport plans [32, 47, 60]. C^2OT [13] further analyzes OT coupling in conditional generation and proposes condition-aware OT to address conditionally skewed source distributions. While these methods clarify the importance of source–target coupling, they typically require costly minibatch optimization and become difficult to estimate reliably in high-dimensional conditional settings. Instead of solving for pairings, Lee et al. [35] learn the coupling by training a target-dependent source encoder jointly with the flow model, regularized toward a Gaussian prior that is also used at inference. Our method shares this learned-coupling view but is designed for conditional generation: the condition-dependent source is used at both training and inference, avoiding the conditional train–test source mismatch analyzed in [13], and only its variance is regularized so that its mean can adapt to the condition.

Condition-dependent source distributions. Several recent works investigate condition-dependent or structured source distributions for conditional flow matching. CPD [28] constructs a condition-dependent prior through a separate regression objective before flow model training, while Ahn et al. [1] learn a source generator from diffusion-inversion trajectories. Other works study source design from geometric, optimization, or reparameterization perspectives [9, 34]. Broader conditional FM frameworks, including WFM [20], Meta Flow Matching [2], and Cocos [16], address related but distinct problems such as conditional parameterization, task adaptation, or distribution-level transport. In contrast, we study jointly learned condition-dependent sources as a mechanism for reducing intrinsic variance and improving FM dynamics in modern T2I systems.

Flow Matching between distributions. Another line of work studies flow matching between different distributions or modalities. Early bridge-based methods [53, 61, 74] primarily focus on related domains such as image-to-image translation. CrossFlow [40] formulates text-to-image generation as cross-modal flow matching, and FlowTok [21] extends this direction to tokenized one-dimensional representations. VAFlow [65] similarly applies cross-modal flow matching to video-to-audio generation. These approaches demonstrate that flow matching can transport across modalities, but their main goal is cross-modal distribution-to-distribution transport. In contrast, we study condition-dependent source learning as a design problem for improving FM dynamics themselves, connecting it to intrinsic variance, stable optimization, and representation-dependent regimes of effectiveness.

L Class-conditional experiment

We additionally conduct class-conditional experiments on ImageNet 256×256 using only the original class labels as conditions. The models are trained with a batch size of 512 for 100K iterations and evaluated using a 50-step Euler sampler without guidance. This budget corresponds to approximately 40 epochs, shorter than typical final ImageNet training schedules; we therefore present it as a controlled comparison under a matched training budget rather than a comparison with published results. As shown in Tab. 10, CSFM outperforms standard FM.

The improvement is smaller than that observed in text-conditioned generation, which is expected. CSFM is most beneficial when the conditioning signal provides sufficiently rich information to align the source and target distributions. This observation is consistent with Sec. 5.3, where CSFM yields larger gains when the target latent space is more strongly correlated with the conditioning signal, as in RAE compared to SD-VAE. It is also consistent with Appx. A.4, where we show that under weak or ambiguous conditioning, the learned source becomes less discriminative across conditions and tends to revert toward a more Gaussian-like baseline. Nevertheless, even in this relatively weak conditioning regime, CSFM still improves over standard FM, demonstrating that condition-dependent source modeling remains beneficial beyond text-conditioned generation.

Table 10: **Class conditional image generation on ImageNet 256×256.**

Method	FID ↓	IS ↑
Standard FM	3.676	189.18
CSFM (Ours)	**3.415**	**194.68**

M Prompt-conditional diversity.

To verify that the improved fidelity does not come from a substantial loss of diversity, we measure prompt-conditional multimodality [72]. For each prompt, we generate 20 images, form 10 random pairs, compute the average pairwise distance in feature space, repeat this five times per prompt, and average over 500 validation prompts. CSFM shows only a modest decrease in multimodality compared to FM, while substantially improving FID. We also report FM with CFG as a reference point, since CFG is known to trade diversity for fidelity.

Table 11: **Prompt-conditional diversity analysis on ImageNet 256×256.** We evaluate whether the fidelity improvement of CSFM comes at the cost of reduced sample diversity. We report multimodality (MModality), computed from pairwise feature distances among multiple samples generated from the same prompt.

Method	FID ↓	MModality ↑
Standard FM	3.036	9.6224 ± 0.3837
Standard FM + CFG	2.801	8.2333 ± 0.3300
CSFM (Ours)	**2.453**	9.2367 ± 0.3702

N Additional target representation analysis

We further evaluate whether the effect of target representation generalizes beyond CSFM by repeating the RAE-versus-SD-VAE comparison with CAR-Flow [9]. We use the same implementation setting as in Appx. E, except that these experiments are conducted with batch size 512. As shown in Fig. 13, CAR-Flow also obtains larger gains in the RAE (DINOv2) representation space than in the SD-VAE space. This supports our conclusion that the target representation geometry is an important factor for condition-dependent source design, beyond the specific choice of CSFM.

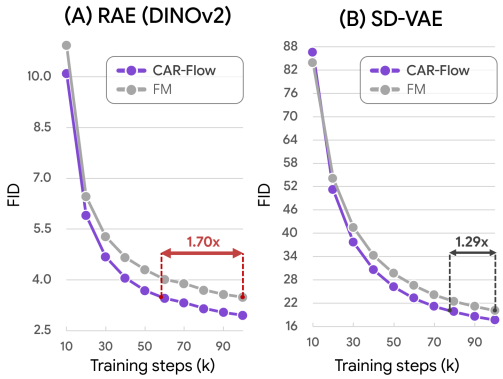

Figure 13: **Target representation analysis with CAR-Flow.** We compare CAR-Flow in SD-VAE and RAE (DINOv2) representation spaces. The larger gain in RAE space indicates that the importance of structured target representations generalizes beyond CSFM.

O Additional ablations

We provide additional ablations on the source regularization, the source generator, and the loss weights. All experiments use the default configuration in Tab. 1, but are trained for 40K iterations

with a batch size of 512; the results are therefore not directly comparable to those in Tab. 1. Across all reasonable variants, CSFM consistently improves over the Gaussian-source baseline.

Source regularization with alignment. As shown in Tab. 12, adding the alignment loss to standard KL regularization improves FID only marginally ($3.693 \rightarrow 3.633$), leaving a clear gap to VarReg with alignment (3.403). Thus, the gain of CSFM cannot be attributed to the alignment loss alone: directional alignment is more effective when the regularizer preserves the mobility of the source mean, and the two components play complementary roles.

Source generator. Our Perceiver-style source generator [36] uses learnable query tokens matching the image-token sequence length, which preserves the token structure while avoiding the large output projection required to map flattened text embeddings (e.g., 77×768 CLIP features) to image latents (e.g., 256×768 RAE latents). We compare it with approximately parameter-matched MLP and self-attention generators, which operate on the text tokens and apply a final projection along the token dimension from 77 to 256 tokens. As shown in Tab. 12, all variants improve over the Gaussian-source baseline, with the Perceiver achieving the best FID among the parameter-matched designs. Varying the capacity of the Perceiver, the 20M and 77M variants perform comparably, while the 130M variant further improves FID, indicating that CSFM remains effective across a range of generator capacities.

Loss weights. The default weights $(\lambda_{\mathrm{VarReg}}, \lambda_{\mathrm{align}}) = (5, 1)$ were selected by placing the weighted auxiliary losses on comparable scales during early training. As shown in Tab. 13, performance is stable across $\lambda_{\mathrm{align}} \in \{0.5, 1.0, 2.0\}$ and $\lambda_{\mathrm{VarReg}} \in \{5.0, 10.0\}$. In contrast, reducing $\lambda_{\mathrm{VarReg}}$ to 1.0 severely degrades performance, consistent with the role of variance regularization in preventing source collapse (Fig. 2 and Tab. 1).

Table 12: **Additional ablations on source regularization and source generator.** All models use the default configuration in Tab. 1 and are trained for 40K iterations with a batch size of 512. The baseline with a fixed Gaussian source is highlighted in gray, and **bold** entries denote the default setting.

Source Distribution	**Reg. Loss**	**Align Loss**	**Source Generator**	FID ↓	CLIP ↑
$\mathcal{N}(0, I)$	✗	✗	–	3.855	0.3346
$p_\phi(X_0 \mid C)$	**VarReg**	**CosSim**	**Perceiver (77M)**	3.403	0.3361
	KL	CosSim	Perceiver (77M)	3.633	0.3358
	KL	✗		3.693	0.3355
	VarReg	CosSim	Perceiver (20M)	3.417	0.3360
			Perceiver (130M)	3.213	0.3364
			MLP (80M)	3.461	0.3362
			Self-Attn (80M)	3.500	0.3368

Table 13: **Sensitivity to loss weights.** All models use the default configuration in Tab. 1 and are trained for 40K iterations with a batch size of 512. The baseline with a fixed Gaussian source is highlighted in gray, and **bold** entries denote the default setting.

Method	$\lambda_{\mathrm{VarReg}}$	λ_{align}	FID ↓	CLIP ↑
Standard FM	–	–	3.855	0.3346
CSFM	**5.0**	**1.0**	3.403	0.3361
	5.0	0.5	3.412	0.3347
	5.0	2.0	3.391	0.3347
	1.0	1.0	20.02	0.3038
	10.0	1.0	3.412	0.3356

P Additional quantitative results

Tab. 14 reports the full benchmark comparison on GenEval [18], and Tab. 15 reports the full benchmark comparison on DPG-Bench [26] between standard FM and CSFM.

Table 14: **Category-wise performance comparison of standard FM and CSFM on GenEval.**

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Standard FM	0.99	0.87	0.63	0.85	0.76	0.53	0.77
CSFM (Ours)	0.98	0.88	0.65	0.90	0.79	0.59	0.80

Table 15: **Category-wise performance comparison of standard FM and CSFM on DPG-Bench.**

Model	Global	Entity	Attribute	Relation	Other	Overall↑
Standard FM	81.46	87.04	85.06	92.80	75.60	78.31
CSFM (Ours)	84.80	88.80	85.93	93.23	76.00	81.11

Q Additional qualitative results

We present qualitative results comparing fixed and learnable source distributions. Fig. 14 and Fig. 15 present qualitative comparisons using the model trained for 100K steps in Sec. 5.1, evaluated on ImageNet-1K [50] validation set, with a 50-step Euler ODE sampler. Fig. 16 further demonstrates model performance on text involving **complex relationships** and **multiple objects**, using the same 50-step Euler ODE sampler. Fig. 17 and Fig. 18 present qualitative comparisons using the scaled UnifiedNextDiT (1.3B). Fig. 19 presents qualitative comparison across the training steps and the NFEs (Number of Function Evaluations), demonstrating faster convergence and straighter flow for CSFM.

R Limitations

Our study uses text-to-image generation as a high-dimensional, large-scale testbed for condition-dependent source learning. Although we evaluate across multiple benchmarks, architectures, and a 1.3B-parameter setting using publicly accessible data and compute, further validation at proprietary frontier-model scale remains future work. We also focus on AutoGuidance rather than standard classifier-free guidance, since backbone-level CFG in CSFM leaves the condition-dependent source unchanged and AutoGuidance is well suited to RAE feature spaces; constructing a fully unconditional branch that also covers the source remains future work. In addition, leaving the source mean unconstrained may reduce robustness under substantial prompt shift. In a preliminary experiment, we evaluate models trained on our captioned ImageNet-1K with short class-name prompts ("An image of {class label}."), whose style is substantially different from the detailed captions used for training (Fig. 12): CSFM obtains an FID of 41.47 compared with 39.58 for the Gaussian-source baseline, although both models degrade considerably. A systematic evaluation under broader prompt and compositional shifts remains future work. Finally, while text-to-image generation provides a representative setting for complex conditional generation, the behavior of CSFM in other domains, such as text-to-audio or text-to-video generation, remains an open direction.

S Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, specifically in the area of generative modeling. While generative models have many potential societal consequences, these are largely well understood in the existing literature. As with other generative image models, the proposed method may generate harmful or inappropriate content depending on the training data and prompts, highlighting the importance of responsible dataset curation and deployment. We do not identify additional societal impacts that require specific discussion beyond these established considerations.

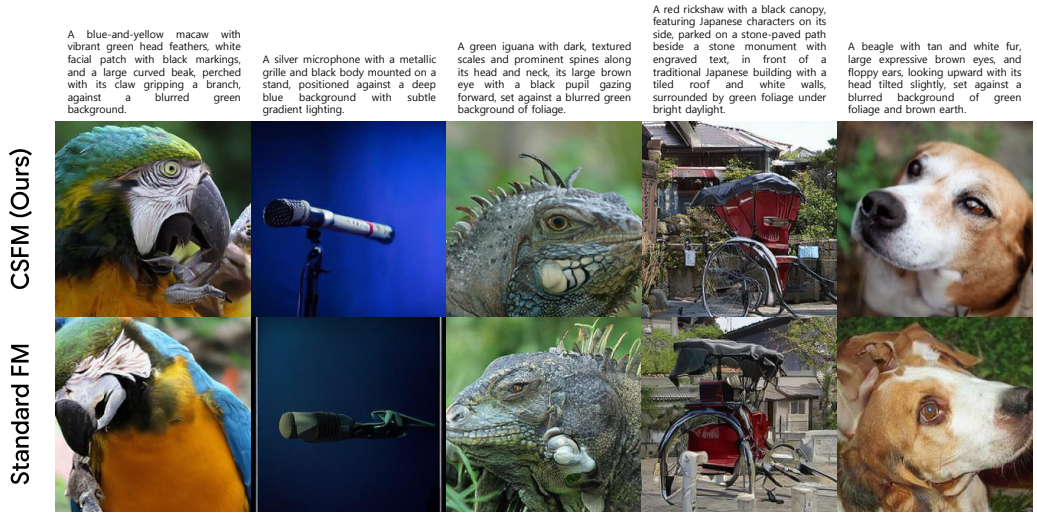

Figure 14: Qualitative comparison between fixed and learned source distributions on ImageNet-1K [50].

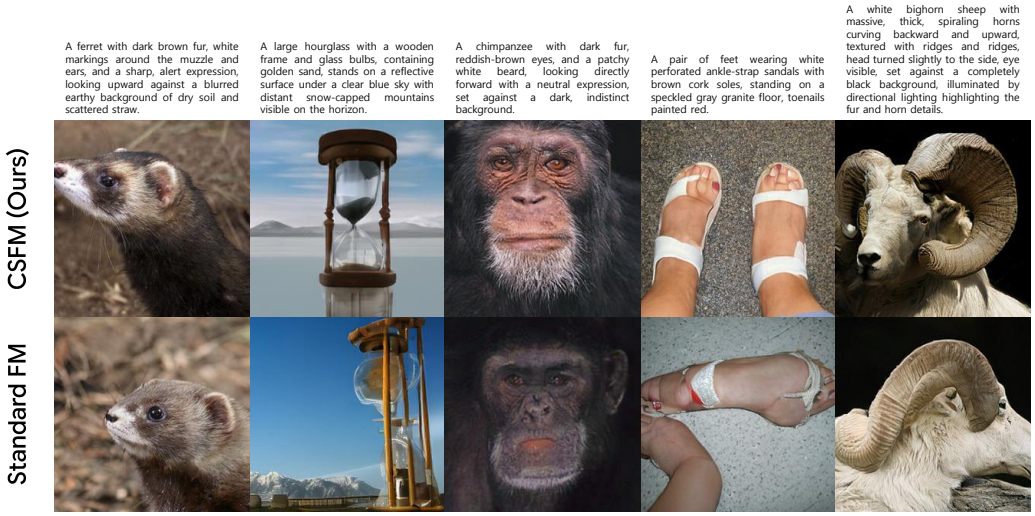

Figure 15: Qualitative comparison between fixed and learned source distributions on ImageNet-1K [50].

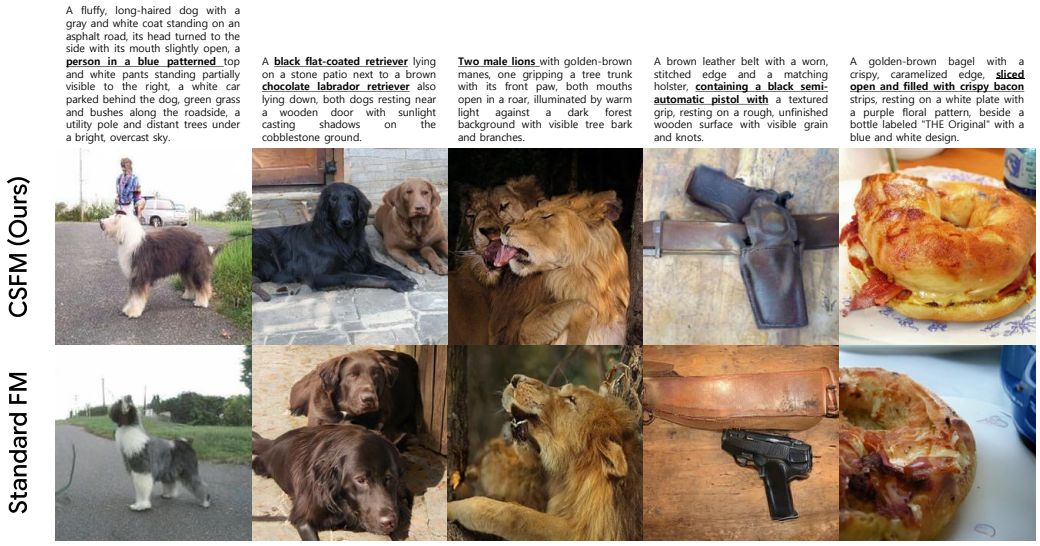

Figure 16: Qualitative comparison between fixed and learned source distributions for prompts with **multiple objects** and **complex relationships**, on ImageNet-1K [50].

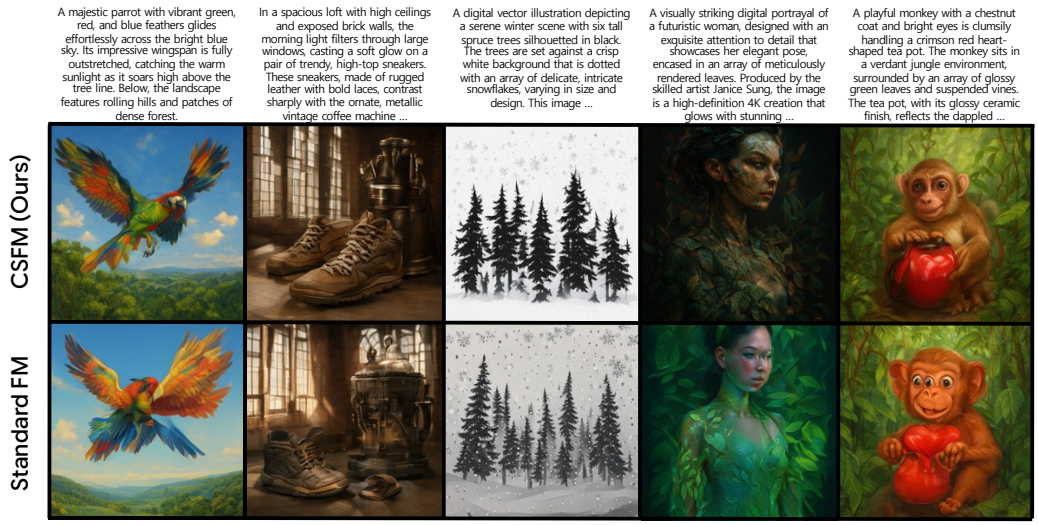

Figure 17: Qualitative comparison between fixed and learned source distributions using Unified-NextDiT (1.3B).



Figure 18: Qualitative comparison between fixed and learned source distributions using Unified-NextDiT (1.3B).

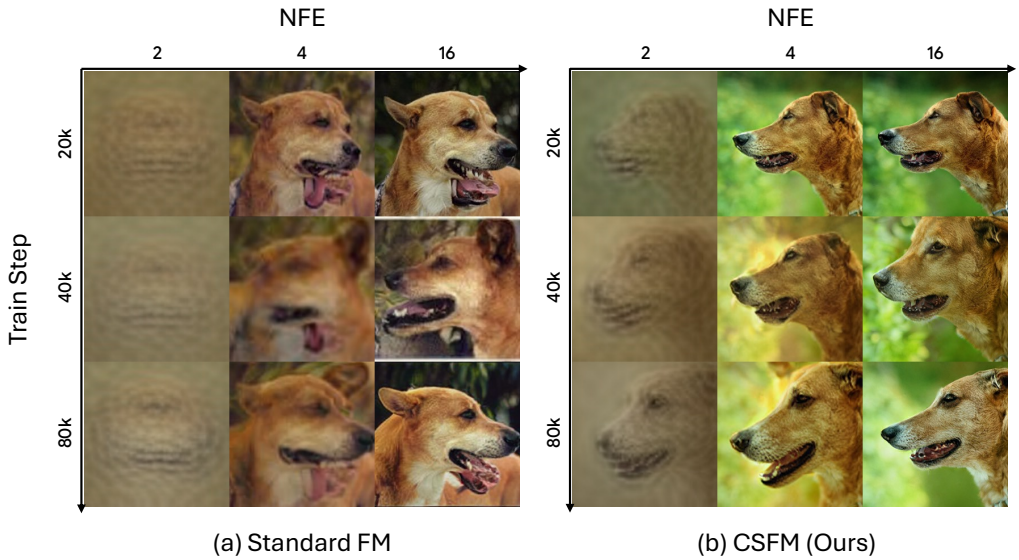

Figure 19: Qualitative comparison between standard FM and CSFM across training steps and NFEs (Number of Function Evaluations) on ImageNet-1K [50].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 clearly state the scope, motivation, and contributions of CSFM. The claims are supported by the proposed method in Sec. 4 and the empirical results in Sec. 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations in Appx. R.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides the bias–variance decomposition with supporting derivations in Appx. C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, model architectures, source generator design, training setup, evaluation metrics, and comparison settings in Sec. 5 and Appx. D / E. These details provide a reasonable path to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is publicly available through the project page linked on the first page, and the captioned ImageNet-1K dataset is publicly released (Appx. H). Details of the experiments and the data are provided in Sec. 5 and Appx. D.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies training iterations, batch sizes, optimizer, learning rates, architectures, evaluation metrics, and benchmark settings in Sec. 5 and Appx. D. It also describes the source generator architecture and objective hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports single-run metric values computed over a large number of samples, due to the computation cost of large-scale text-to-image models.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper states the cluster for training and evaluation, and time of execution in Appx. D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the potential impacts in Appx. S.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The submission does not release a model with misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite the original assets in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the captions of our captioned ImageNet-1K dataset under the MIT license. Its construction is documented in Appx. H; the images themselves remain subject to the original ImageNet terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLMs to develop any methods presented in this work. We used LLMs only for writing purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.