Rethinking Vector Field Learning for Generative Segmentation

Chaoyang Wang[1] **Yaobo Liang**[1] **Boci Peng**[1] **Fan Duan**[2] **Yunhai Tong**[1]
[1]State Key Lab of General AI, School of Intelligence Science and Technology, Peking University
[2]Tsinghua University
cywang@stu.pku.edu.cn

Abstract

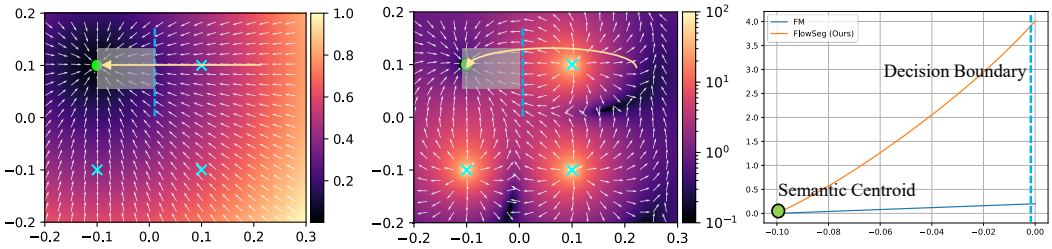
Taming diffusion models for generative segmentation has attracted increasing attention. While existing approaches primarily focus on architectural tweaks or training heuristics, there remains a limited understanding of the intrinsic mismatch between continuous flow matching objectives and discrete perception tasks. In this work, we revisit diffusion segmentation from the perspective of vector field learning. We identify two key limitations of the commonly used flow matching objective: gradient vanishing and trajectory traversing, which result in slow convergence and poor class separation. To tackle these issues, we propose a principled vector field reshaping strategy that augments the learned velocity field with a detached distance-aware correction term. This correction introduces both attractive and repulsive interactions, enhancing gradient magnitudes near centroids while preserving the original diffusion training framework. Furthermore, we design a computationally efficient, quasi-random centroid encoding scheme inspired by Kronecker sequences, which integrates seamlessly with an end-to-end pixel neural field framework for pixel-level semantic alignment. Extensive experiments consistently demonstrate significant improvements over vanilla flow matching approaches, substantially narrowing the performance gap between generative segmentation and strong discriminative specialists. Project Page: https://wang-chaoyang.github.io/project/flowseg.

1 Introduction

Diffusion models [34, 35, 16] have emerged as a leading approach in visual generation, excelling at modeling complex data distributions. Their effectiveness in high-quality image synthesis [32, 30, 12], video generation [4, 1, 15] and image editing [5] has motivated extensions to various downstream tasks. While initially proposed for visual generation, recent works [44, 38, 37] have begun applying diffusion frameworks to perception tasks, aiming to unify generation and understanding within a common paradigm. Among these tasks, segmentation is distinctive in that it assigns each pixel a discrete semantic label [29, 8, 33, 48, 11, 10], producing categorical rather than continuous outputs. This poses a key challenge: diffusion models operate in continuous spaces, while segmentation requires inherently discrete predictions. This gap introduces significant challenges in task formulation and optimization.

To adapt diffusion models for segmentation and utilize their pretrained priors, existing methods fall into three main categories, as shown in Fig. 2. 1) Employing visual foundation features and adding lightweight diffusion modules to refine masks [39, 14, 13]. 2) Using diffusion models as feature extractors with segmentation heads [49, 42, 20, 17, 36, 3, 18]. 3) Treating segmentation as end-to-end image translation, encoding masks as three-channel pseudo-color images [2, 21, 37, 38, 31, 24]. Beyond model design, interestingly, semantic segmentation remains a challenging and underexplored task for diffusion-based models, primarily due to two factors: 1) the mismatch between the inherent

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



(a) Vanilla FM $\nabla_x L \propto (x - \mu)$ (b) FlowSeg (Ours) $\nabla_x L \propto (x - \mu + \nabla\Phi)$ (c) Gradient Norm

Figure 1: **Visualization of gradient vanishing and trajectory traversing in generative segmentation (x-prediction).** (a) Vanilla flow matching suffers from vanishing gradients near semantic centroids μ, resulting in slow and non-discriminative trajectories that may traverse proximal neighborhoods of competing centroids and cause false predictions. (b) FlowSeg (ours) introduces a potential function Φ to enhance gradients around target centroids and enforce repulsion from non-targets, enabling faster convergence and more discriminative, deflected trajectories. The $x - \mu$ term maintains convergence near outer boundaries. (c) Gradient norm from centroid to decision boundary (gray regions in (a),(b)): Our method maintains strong gradients, whereas vanilla flow matching gradients nearly vanish. Yellow curves and blue dashed lines denote the predicted trajectory and decision boundary. The green dot marks the target centroid, and blue crosses indicate irrelevant categories.

stochasticity of generative models and the deterministic demands of semantic segmentation, and 2) the typically large number of classes involved. Recent works [38, 21] address the first challenge by introducing deterministic flows, yet the issue of high class cardinality persists. To maintain tractability, existing approaches often simplify the problem setting. For instance, some methods reduce the class space by merging similar categories [43], or reformulate the task as a series of referring image segmentation problems [14] to break it into simpler subproblems. These findings indicate that stochasticity and high class cardinality are fundamental obstacles for diffusion segmentation, suggesting that the core difficulty lies not in model capacity, but in the optimization dynamics inherent to the framework.

In this paper, we analyze the issue from an optimization dynamics perspective and identify two key problems: *gradient vanishing* and *trajectory traversing*, as shown in Fig. 1. By examining the gradients induced by standard flow matching [25, 26], we observe that the optimization signal is directly scaled by the distance between predictions and semantic centroids. As class centroids reside in a bounded low-dimensional space, the gradients are inherently limited and decay rapidly as predictions approach the target, ultimately vanishing near convergence and causing optimization to stall. Furthermore, the regression-based approach only provides attractive forces toward the ground-truth class, lacking explicit repulsion from competing centroids. This combination of weak gradients and absent inter-class repulsion results in slow convergence and poor semantic separation, ultimately degrading segmentation performance.

To overcome these limitations, we reshape the vector field by introducing a distance-aware correction term to the original velocity, enabling both attractive and repulsive interactions and sustaining non-vanishing gradients near centroids. This enhances class separation and accelerates convergence. Additionally, we propose a quasi-random centroid encoding strategy inspired by Kronecker sequences, which is computationally efficient. Combined with an end-to-end pixel neural field framework, our design aligns the diffusion model with pixel-level tasks.

Our main contributions are summarized as follows: 1) We analyze the slow convergence and suboptimal performance in diffusion segmentation, attributing them to gradient vanishing and trajectory traversing caused by the vanilla flow matching objective. 2) We propose a novel vector field reshaping strategy that augments target velocity with a corrective term, preserving gradient magnitude and introducing explicit repulsion to improve semantic separation. 3) We employ a pixel neural field framework for end-to-end training and introduce a scalable, quasi-random centroid encoding scheme that ensures balanced inter-class geometry without extra optimization. 4) Extensive experiments show consistent and significant improvements, substantially narrowing the gap between generative and strong discriminative approaches.

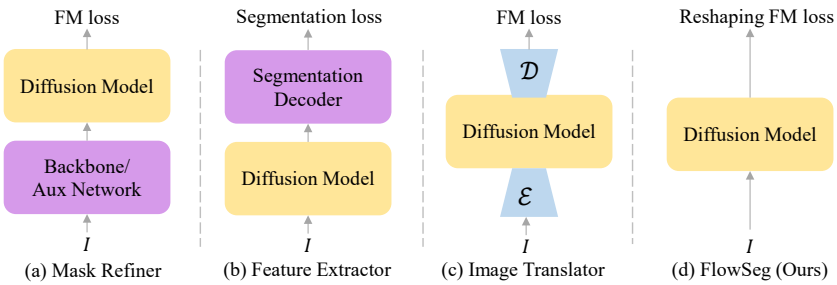


Figure 2: **Comparison of diffusion segmentation paradigms.** (a) Diffusion models are used primarily for mask refinement, relying on an external backbone for feature extraction or auxiliary networks for coarse segmentation. (b) Diffusion models serve as the backbone, followed by a dedicated segmentation head. (c) The segmentation task is formulated as image-to-mask translation without auxiliary networks, yet still depends on a pretrained VAE. (d) FlowSeg (ours) performs pixel-level end-to-end training without additional auxiliary modules, and rectifies vanilla flow matching by reshaping the underlying vector field for better optimization. Noise is omitted for simplicity.

2 Related Work

2.1 Diffusion and Flow Model

Diffusion models [16] and flow models [25, 27] parameterize data generation as a time-dependent dynamical system that transports a simple reference distribution to a target data distribution. They enable high-fidelity synthesis across diverse domains, including image generation [32, 30, 12], video generation [4, 1, 15], image editing [5], etc. Early diffusion models typically use stochastic differential equations (SDEs) to learn score or noise estimates under a predefined Gaussian corruption process. In contrast, flow models [27, 28] directly learn a deterministic velocity field that defines a straight probability path, enabling more efficient sampling and greater flexibility in distribution transport.

To reduce computational overhead, conventional approaches [32] predominantly operate within latent spaces compressed by Variational Autoencoders (VAEs) [19]. However, the inherent information loss in VAE compression often hinders tasks that require pixel-level precision and direct spatial alignment. To address this, recent pixel diffusion models [46, 22, 40] have emerged, leveraging large patch decoding to maintain scalability without sacrificing fine-grained details. Despite differences in parameterization, these approaches share a common principle: generative modeling is reduced to learning a *continuous* vector field that aligns probability trajectories between distributions, typically via regression-based objectives, which poses a challenge for extending to *discrete* perception tasks.

2.2 Generative Segmentation

Visual segmentation aims to partition an image into semantically coherent regions with discrete class labels. While conventional methods [29, 8, 33, 48, 11, 10] typically frame segmentation as a discriminative dense classification task, generative formulations have recently gained traction for their superior uncertainty modeling and structured prediction capabilities [2, 23].

Existing diffusion segmentation methods generally follow three paradigms: 1) Refinement-based, which employ diffusion modules to iteratively polish masks generated by foundation models [39, 14, 13]; 2) Feature-based, which utilize diffusion backbones as powerful frozen extractors for task-specific segmentation heads [49, 42, 20, 17, 36, 3, 18]; 3) Unified-generative, which cast mask prediction as an end-to-end image translation process [2, 21, 37, 38, 31, 24].

In the unified-generative formulation, segmentation masks are typically embedded into a low-dimensional continuous manifold to align with the continuous nature of generative dynamics. While such embedding is essential for generative modeling, simply adopting vanilla regression-based flow objectives fails to account for the discrete nature of semantic labels. This leads to an optimization mismatch where the vector field is learned as a generic density estimator, overlooking the categorical constraints inherent in segmentation.

3 Optimization Dynamics Analysis

In this section, we provide a formal analysis of why the standard Mean Squared Error (MSE) objective, commonly used in Flow Matching for generative tasks, leads to suboptimal optimization dynamics and performance bottlenecks when applied to generative segmentation.

3.1 Preliminary

Let $I \in \mathbb{R}^{H\times W\times 3}$ denote an input image. We define the segmentation task as a flow between a source distribution $p_0(x_0|I)$ and a target distribution $p_1(x_1)$. To eliminate the randomness of Gaussian noise and leverage the deterministic nature of segmentation, we define x_0 as a feature-dependent distribution (e.g., the input image itself or encoded image features) and x_1 as the target mask representation.

In a categorical or multi-channel segmentation setting, each pixel or region is mapped to a specific target centroid within a bounded continuous space (e.g., $\mathcal{C} \subset [-1, 1]^d$). Let $\mathcal{P} = \{\mu_1, \mu_2, \ldots, \mu_N\}$ be the set of target centroids representing N distinct categories or instances. In Rectified Flow [27], the probability path x_t is constructed as follows:

$$x_t = (1 - t)x_0 + tx_1, \quad t \in [0, 1] \tag{1}$$

The corresponding ground-truth velocity (vector field) is:

$$v_{gt}(x_t) = \frac{dx_t}{dt} = x_1 - x_0 \tag{2}$$

The standard training objective is to learn a velocity field $v_\theta(x_t, t, I)$ parameterized by a neural network, which approximates the ground-truth via the MSE loss:

$$\mathcal{L}_{FM} = \mathbb{E}_{t,x_0,x_1} \left\| v_\theta(x_t, t, I) - v_{gt} \right\|^2 \tag{3}$$

3.2 Gradient Analysis

To understand the optimization challenges, we analyze the gradient of $\mathcal{L}_{FM}$ with respect to the network's prediction. The velocity v_θ implicitly defines a predicted target $\hat{x}_1$, which can be estimated as:

$$\hat{x}_1 = x_t + (1 - t)v_\theta \tag{4}$$

Substituting v_θ and v_{gt} into the loss function, we can rewrite the objective in terms of the distance between the predicted state and the target centroid:

$$\mathcal{L}_{FM} \propto \left\| \frac{\hat{x}_1 - x_t}{1 - t} - \frac{x_1 - x_t}{1 - t} \right\|^2 = \frac{1}{(1 - t)^2} \left\| \hat{x}_1 - x_1 \right\|^2 \tag{5}$$

By examining the gradient of this loss with respect to the predicted target $\hat{x}_1$, several critical issues emerge regarding the optimization dynamics:

Gradient Vanishing Near Centroids. The gradient magnitude is directly proportional to the Euclidean distance between the prediction and the target for fixed t:

$$\nabla_{\hat{x}_1} \mathcal{L}_{FM} \propto (\hat{x}_1 - x_1) \tag{6}$$

As the prediction $\hat{x}_1$ approaches the target centroid $x_1 \in \mathcal{P}$, the gradient vanishes. In the late stages of training, the model lacks sufficient driving force to precisely converge to the exact centroid coordinates, leading to blurry segmentation boundaries.

Limited Signal Scale in Bounded Space. Given that the centroids $\mathcal{P}$ are typically embedded in a compact space such as $[-1, 1]^d$, the maximum possible distance $\|\hat{x}_1 - x_1\|$ is intrinsically small. For tasks with high category cardinality N, the centroids are densely packed, and the MSE-based gradient provides a very weak signal for the model to distinguish between adjacent categorical anchors.

Absence of Repulsive Forces. The MSE loss is strictly a *unimodal attraction* objective. It solely penalizes the distance to the target centroid and remains agnostic to competing centroids. On the one hand, from the perspective of classical contrastive learning, a robust classifier should not only pull the prediction toward the positive category but also push it away from negative ones. On the other hand, from the perspective of flow matching, without a repulsive mechanism, target-bound trajectories may inadvertently *traverse* the proximal neighborhoods of competing classes, inducing semantic ambiguity during the generative process.

4 Method

In this section, we detail the proposed framework to rectify the optimization dynamics of generative segmentation. First, we introduce a deterministic, quasi-random scheme for category encoding. Then, we present the vector field reshaping mechanism, which is designed to introduce discriminative forces into the generative training objective. Finally, we propose using pixel neural field decoding for end-to-end generative segmentation training.

4.1 Category Encoding

To map N semantic categories into the bounded 3D color space $\mathcal{C} \subset [-1, 1]^3$, we employ a Kronecker-style sequence, which is a quasi-random sequence characterized by determinism and low computational cost.

Let $\mathcal{V} = \{\sqrt{2}, \sqrt{3}, \sqrt{5}\}$ be a set of square roots of the first three primes. These values are linearly independent over the field of rational numbers $\mathbb{Q}$, ensuring that the generated sequence does not collapse onto lower-dimensional manifolds or exhibit periodic correlations. For each category index $k \in \{0, \dots, N-1\}$, the centroid μ_k is initially computed as:

$$\hat{c}_k = (k \cdot \mathcal{V}) \pmod 1 \tag{7}$$

The resulting coordinates are then normalized and stretched to occupy the full volume of the $[-1, 1]^3$ cube:

$$\mu_k = 2 \cdot \text{Norm}(\hat{c}_k) - 1 \tag{8}$$

This approach ensures a relatively uniform distribution of centroids with high minimum inter-point distances, providing a stable and deterministic geometric basis for the vector field learning.

4.2 Vector Field Reshaping

As analyzed in Sec. 3, the vanilla flow matching objective fails to provide repulsive forces, leading to path traversing and gradient vanishing. To address this, we reshape the flow matching objective by augmenting the ground-truth velocity v_{gt} with a discriminative rectification term.

Potential-based Vector Rectification. To instill discriminative awareness into the flow field, we construct a potential field Φ over the centroid space. Let $\hat{x}_1$ denote the estimated prediction, and $d_k = \|\hat{x}_1 - \mu_k\|^2$ be the squared Euclidean distance to target centroid $\mu_k \in \mathcal{P}$. We introduce a transformation operator $\mathcal{T}(\cdot)$ to map these geometric distances into a semantic embedding space. In our primary formulation, we define this operator as $\mathcal{T}(d) = -\log(d + \epsilon)$, where ϵ is to avoid zero-division. $\mathcal{T}$ remains flexible for various distance-warping strategies.

Distances are converted into a soft-assignment distribution p via a temperature-scaled softmax over the transformed distances:

$$p_k = \frac{\exp(\mathcal{T}(d_k)/\tau)}{\sum_j \exp(\mathcal{T}(d_j)/\tau)} \tag{9}$$

where τ is a temperature hyperparameter controlling the sharpness of the categorical assignment. To guide the optimization, we define the discriminative potential Φ as the divergence between the current assignment p and the ground-truth one-hot distribution y:

$$\Phi(\hat{x}_1, y) = -\sum_{k=1}^{N} y_k \log p_k \tag{10}$$

This potential reaches its global minimum when the prediction $\hat{x}_1$ perfectly aligns with the target centroid μ_{gt} while maximizing its separation from competitors. The rectification term $\nabla\Phi$ is obtained by computing the gradient of the potential with respect to the prediction $\hat{x}_1$. Utilizing the property $\mathcal{T}'(d) = -(d + \epsilon)^{-1}$, the gradient is formulated as:

$$\nabla\Phi = \frac{\partial\Phi}{\partial\hat{x}_1} = \frac{2}{\tau}\sum_{k=1}^{N}(y_k - p_k)\frac{\hat{x}_1 - \mu_k}{d_k + \epsilon} \tag{11}$$

Objective Reformulation. The reshaped target velocity $\tilde{v}_t$ is defined by integrating the rectification term into the original velocity:

$$\tilde{v}_t = v_t - \nabla\Phi \tag{12}$$

To maintain the stability of the probability path and prevent the network from directly optimizing the potential field, we apply the stop-gradient operator to the reshaped target. The final training objective is defined as:

$$\mathcal{L}_{res} = \mathbb{E}_{t,x_0,x_1} \left\| v_\theta(x_t, t, I) - \text{sg}[\tilde{v}_t] \right\|^2 \tag{13}$$

where $\text{sg}[\cdot]$ denotes the detach operation.

Gradient Dynamics Comparison. To further elucidate the impact of Flow Reshaping, we compare the gradients of the standard Flow Matching loss and our proposed objective. For the sake of clarity, let $g_\theta = \nabla_\theta v_\theta$ denote the Jacobian of the network with respect to its parameters.

In the standard Flow Matching framework, the parameter update is driven by:

$$\nabla_\theta \mathcal{L}_{FM} = g_\theta^\top \left(v_\theta - (x_1 - x_0) \right) \tag{14}$$

As v_θ approaches the constant target $v_{gt} = x_1 - x_0$, the error signal diminishes. This signal is purely reconstructive, providing no information regarding the proximity of the predicted flow to incorrect categorical centroids.

In contrast, our reshaped objective $\mathcal{L}_{res}$ yields the following gradient:

$$\nabla_\theta \mathcal{L}_{res} = g_\theta^\top \left(v_\theta - (x_1 - x_0 - \text{sg}[\nabla \Phi]) \right) \tag{15}$$

By decomposing the term $\nabla \Phi$, we observe that a discriminative bias shifts the effective target for the network:

$$\text{Effective Target} = \underbrace{(x_1 - x_0)}_{\text{Reconstructive}} + \underbrace{\frac{2}{\tau} \sum_{k=1}^{N} (p_k - y_k) \frac{\hat{x}_1 - \mu_k}{\left\| \hat{x}_1 - \mu_k \right\|^2 + \epsilon}}_{\text{Discriminative Correction}} \tag{16}$$

4.3 End-to-End Pixel Decoding

Conventional VAE-based latent spaces are often ill-suited for high-precision segmentation due to *manifold distortion* and *optimization decoupling*. The former introduces latent artifacts that disrupt pixel-level details, while the latter prevents the diffusion process from aligning directly with segmentation objectives. Moreover, the high memory overhead of VAEs makes full end-to-end optimization computationally infeasible.

To circumvent these limitations, following PixNerd [40], we propose using pixel neural field decoding for end-to-end generative segmentation training. Instead of decoding a patch feature $\mathbf{X}^n \in \mathbb{R}^D$ through a simplistic linear projection, we treat each patch as a continuous neural field. The Transformer backbone serves as a parameter generator, predicting the weights of a localized Multi-Layer Perceptron (MLP). For the n-th patch, the network dynamically predicts the weight matrices $\{\mathbf{W}_1^n, \mathbf{W}_2^n\}$ of a lightweight MLP from the patch feature. These weights are generated via a projection and subsequently normalized to ensure numerical stability, yielding the normalized weights $\bar{\mathbf{W}}_1^n$ and $\bar{\mathbf{W}}_2^n$:

$$\mathbf{W}_1^n, \mathbf{W}_2^n = \text{Linear}(\mathbf{X}^n)$$

To query the velocity field at any spatial coordinate (i, j) within the patch, we construct an input context $\mathbf{h}_{i,j}$ by concatenating the Discrete Cosine Transform (DCT) positional encoding with the noisy pixel state $\mathbf{x}_t(i, j)$:

$$\mathbf{h}_{i,j} = \text{Concat}([\text{DCT}(i, j), x_t(i, j)])$$

The final pixel-wise velocity $v^n(i, j)$ is then directly decoded by applying the dynamically generated MLP:

$$v^n(i, j) = \bar{\mathbf{W}}_2^n \sigma(\bar{\mathbf{W}}_1^n \mathbf{h}_{i,j})$$

where σ denotes the SiLU activation function.

5 Experiment

In this section, we first detail our experimental settings and benchmark our approach against baseline methods both quantitatively and qualitatively. Finally, we present comprehensive ablation studies dissecting our macroscopic modeling choices, objective formulations, operator dynamics, and state representations.

Table 1: **Comparisons on ADE20K and COCO-Stuff dataset.** † indicates that it performs referring image segmentation, which requires predicting a separate mask for each semantic category. LSTI and (·) indicate large-scale text-image datasets and the initialized weight, respectively. **Bold** and underline indicate the first and second best entries.

(a) Performance on ADE20K dataset

Method	Backbone	Pretrain Data	mIoU
Discriminative Specialists			
DeeplabV3+	ResNet101	IN-1k	44.1
SegFormer	MiT-B2	IN-1k	46.5
MaskFormer	Swin-T	IN-1k	<u>46.7</u>
Diffusion Models			
InstructDiffusion†	(SD1.5)	LSTI	33.6
PixWizard	(Lumina-Next-T2I)	LSTI	32.8
FlowSeg (Ours)	(PixNerd)	IN-1k	**47.1**

(b) Performance on COCO-Stuff dataset

Method	Backbone	Pretrain Data	mIoU
Discriminative Specialists			
DeeplabV3+	ResNet50	IN-1k	38.4
OCRNet	HRNet-W48	IN-1k	42.3
SegFormer	MiT-B2	IN-1k	<u>44.6</u>
Diffusion Models			
SymmFlow	(SD2.1)	LSTI	39.6
FlowSeg (Ours)	(PixNerd)	IN-1k	**44.9**

5.1 Experimental Settings

Dataset. We evaluate our approach on two high-cardinality datasets: ADE20K [50] and COCO-Stuff [6]. ADE20K is a scene parsing dataset covering 150 fine-grained semantic concepts consisting of 20k images. COCO-Stuff covers 171 labels and consists of 164k images.

Implementation. Our model is initialized with PixNerd [40] weights. All images and segmentation masks are resized to a uniform resolution. Training proceeds in two progressive stages, using the AdamW optimizer and Representation Alignment (REPA) [45] to accelerate convergence. Stage 1 uses a 256×256 resolution with a batch size of 64, followed by Stage 2 at 512×512 with a batch size of 16. The default temperature is 1.0. In line with standard practice, binary masks are applied to filter out unlabeled regions, which are omitted from our notation for brevity.

Baselines. We consider discriminative specialists and diffusion models. For the former, we include DeeplabV3+ [9], OCRNet [47], SegFormer [41] and MaskFormer [11]. For the latter, we include InstructDiffusion [14], PixWizard [24] and SymmFlow [7].

5.2 Main Results

Quantitative Results. Tab. 1a and Tab. 1b benchmark FlowSeg against discriminative specialists and diffusion-based models. Existing diffusion paradigms lag significantly behind discriminative baselines, even when leveraging the extensive priors of large-scale pretrained models like Stable Diffusion [32]. This performance gap corroborates our analysis in Sec. 3 regarding the optimization bottlenecks inherent to generative segmentation. Specifically, on ADE20K, InstructDiffusion and PixWizard trail DeepLabV3+ by over 10 mIoU. Similarly, on COCO-Stuff, SymmFlow underperforms SegFormer by 5 mIoU despite utilizing SD2.1 weights. In contrast, empowered by the reshaped vector field, FlowSeg achieves 47.1 mIoU on ADE20K and 44.9 mIoU on COCO-Stuff. Notably, FlowSeg successfully surpasses strong discriminative specialists solely relying on IN-1k pretraining.

Qualitative Results. Fig. 3 demonstrates FlowSeg's robust segmentation capabilities across diverse and complex scenarios, effectively resolving varying object scales in both indoor and outdoor environments. Furthermore, we contrast FlowSeg with SymmFlow, which maps the joint distribution of images and Gaussian noise to segmentation masks. As shown in Fig. 4a, SymmFlow's stochastic formulation fails to yield deterministic predictions, leading to inconsistent outputs. Conversely, FlowSeg strictly enforces deterministic transport, yielding qualitatively superior and highly consistent semantic boundaries.

5.3 Ablation Studies

Macroscopic Modeling. We validate our macroscopic design in Fig. 4 and Fig. 5. Unlike SymmFlow, which relies on Gaussian noise and yields unstable predictions across random seeds in Fig. 4a, FlowSeg strictly leverages a deterministic flow to meet semantic segmentation requirements. Furthermore, traditional VAE-based methods struggle with pixel-wise precision due to latent space distortion. In contrast, our pixel-level end-to-end modeling reliably distinguishes fine-grained semantics in Fig. 4b. This architecture also guarantees superior convergence. Fig. 5a demonstrates FlowSeg

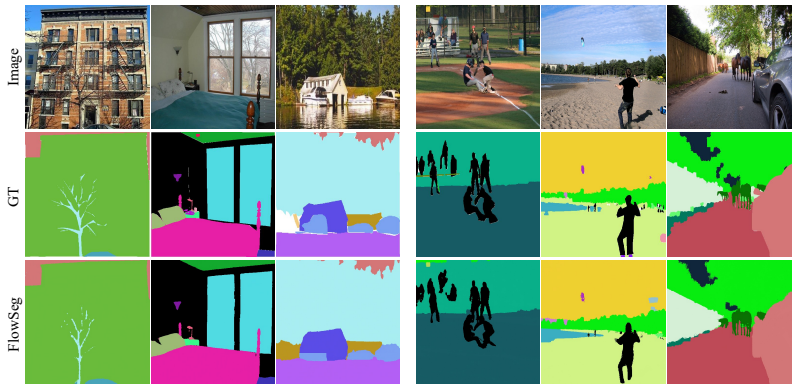


Figure 3: Visualization of segmentation results on (a) ADE20K and (b) COCO-Stuff datasets. Color white in the ground truth (GT) denotes the ignored regions. As ADE20K and COCO-Stuff datasets have different category cardinality, the same color between (a) and (b) does not necessarily represent the same semantic category.

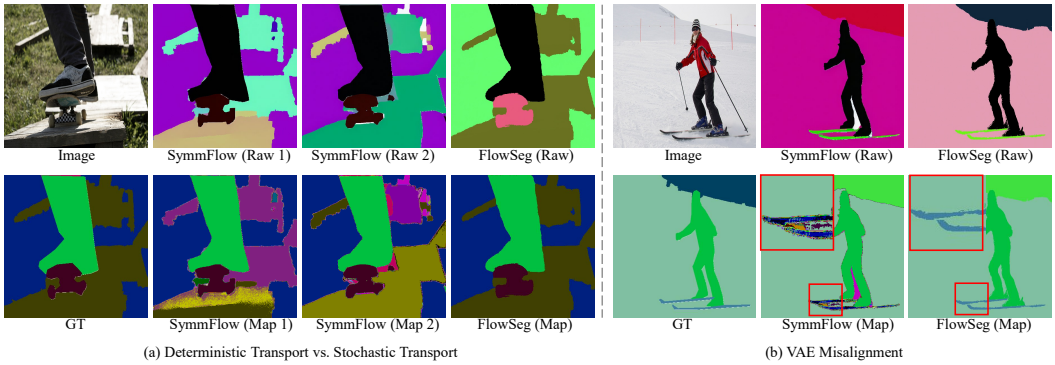
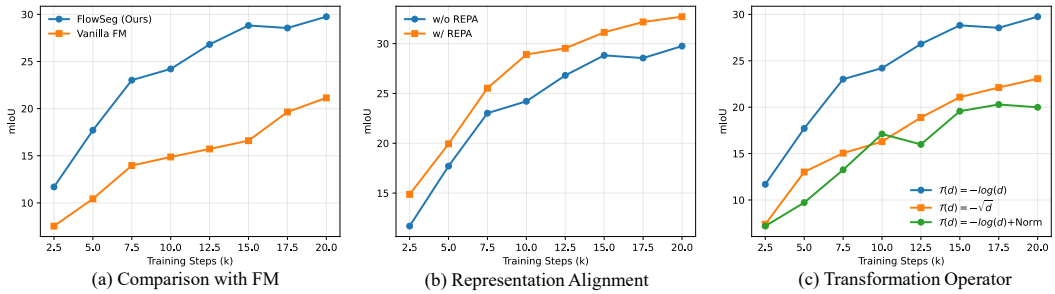


Figure 4: **Visual comparisons between FlowSeg (ours) and SymmFlow (Baseline).** The diffusion model first predicts pseudo-masks (**Raw**), then maps them to the nearest semantic centroids to obtain the final masks (**Map**). (a) Comparison between deterministic (ours) and stochastic modeling: SymmFlow's predictions vary with random seeds, while ours remain consistent. (b) VAE-based latent space modeling produces masks with similar colors that may not correspond to the correct semantic categories, due to imperfect alignment with pixel-level centroids.

converges significantly faster than vanilla flow matching from the outset. Incorporating REPA further accelerates this process, boosting mIoU from 29.8 to 32.7 in Fig. 5b.

Training Objectives. Tab. 2a and Tab. 2b detail our objective formulation. First, excluding unlabeled pixels effectively prevents semantically ambiguous areas from disrupting optimization, improving the v-prediction baseline from 28.15 to 29.75 mIoU in Tab. 2a. We also observe a modest performance gap between v-prediction and x-prediction. Second, Tab. 2b reveals that the failure of vanilla x-prediction is intrinsic to transport field geometry. Explicit time compensation via $(1-t)^2$ fails to rescue vanilla flow matching, dropping performance from 22.05 to 18.30 mIoU. Conversely, our reshape field substantially boosts x-prediction to 29.06 mIoU, peaking at 30.68 mIoU with time compensation.

Operator Dynamics. The geometric reshaping strictly relies on the transformation operator $\mathcal{T}$, which dictates the proxy gradient by mapping squared Euclidean distances d to class posteriors via softmax. Analyzed in Tab. 2d and Fig. 5c, an effective operator must drive the target logit to strongly dominate, suppressing incorrect categories by creating a sharp probability peak and a sufficient margin. As $d \to 0$, the logarithmic term $-\log(d) \to +\infty$, providing a natural singularity with a manageable gradient proportional to $1/d$. This avoids the explosive instability of the hyper-singular $1/(d+\epsilon)$, which collapses to 5.46 mIoU. Conversely, milder operators like $1/\sqrt{d}$ (23.44 mIoU) exhibit sluggish



(a) Comparison with FM (b) Representation Alignment (c) Transformation Operator

Figure 5: Convergence comparison of different training recipes. (a) FlowSeg vs. vanilla flow matching. (b) Training w/ REPA vs. w/o REPA, (c) Different transformation operators $\mathcal{T}$.

Table 2: **Ablations on training, representations, and operators.** (a) Mask excludes unlabeled regions (defaults use v-pred). (b) T-comp scales x-pred loss by $(1-t)^2$ to offset time dependence. (c) Concat conditions images with padded maps (Gaussian/Zero initialized) under vanilla flow matching. K denotes category cardinality.

(a) Design choices

Pred.	Mask	mIoU
v-pred		28.15
v-pred	✓	29.75
x-pred	✓	29.06

(b) Time compensation

Reshape Field	T-comp.	mIoU
		22.05
	✓	18.30
✓		29.06
✓	✓	30.68

(c) Dimension ablation

Dim	Concat	Init	mIoU
3	-	-	20.60
3	✓	Gauss	5.03
3	✓	Zero	17.28
K	✓	Gauss	0.29
K	✓	Zero	1.02

(d) Operator ablation

Oper.	mIoU
$-\log(d)$	29.75
$1/(d+\epsilon)$	5.46
$1/\sqrt{d}$	23.44
$-\log(d) - \sqrt{d}$	30.73

convergence due to weak far-field shaping. The optimal formulation, $-\log(d) - \sqrt{d}$ (30.73 mIoU), seamlessly bridges near-centroid singularity with a far-field distance penalty. Ultimately, a robust operator must simultaneously enforce near-centroid attraction, wrong-centroid suppression, and stable global gradients.

Representation and Initialization. Tab. 2c evaluates the flow state space. The compact 3-channel cube drastically outperforms the K-channel one-hot embedding, yielding 17.28 versus 1.02 mIoU under zero initialization. One-hot targets reside on sparse simplex vertices. When paired with a Gaussian source, the noise spans the entire $\mathbb{R}^K$ space. For high cardinality (e.g., $K = 150$), this injects massive off-simplex perturbations along semantically meaningless directions, forcing the model into irrelevant high-dimensional denoising. Since segmentation demands deterministic dense prediction rather than inherently stochastic generation, these unconstrained perturbations severely disrupt the transport mapping. Consequently, zero initialization strictly surpasses Gaussian initialization. While Gaussian noise also degrades the 3-channel variant (dropping from 17.28 to 5.03 mIoU), the lower-dimensional state space reduces the expected perturbation magnitude. In contrast, high-dimensional K-channel Gaussian noise thoroughly derails optimization by dragging trajectories into non-semantic spaces.

6 Conclusion

This paper explores generative segmentation via flow matching, particularly in high-cardinality segmentation scenarios. We identified two key obstacles to previous methods: gradient vanishing and trajectory traversing. However, these problems are further obscured and entangled by classical latent-space modeling. To tackle these issues, we propose an end-to-end generative segmentation pipeline based on a pixel neural field. Then, we propose a novel vector field reshaping approach that simultaneously solves both problems while maintaining a high gradient near semantic centroids and introducing repulsion in the velocity field. Empirical studies and comprehensive ablations demonstrate significant improvements in convergence speed and final performance over classic flow-matching training recipes, and they closely approach strong discriminative specialists. We hope our research can inspire the community to rethink generative segmentation.

Acknowledgment. This work was supported by NSFC (Grant No.42271426) and Shenzhen Science and Technology Innovation Project (Grant No.2025110105).

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[3] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022.

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[7] Francisco Caetano, Christiaan Viviers, Peter HN De With, and Fons van der Sommen. Symmetrical flow matching: Unified image generation, segmentation, and classification with score-based generative models. In *AAAI*, 2026.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[11] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

[12] 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. *arXiv preprint arXiv:2403.03206*, 2024.

[13] Yue Fan, Yongqin Xian, Xiaohua Zhai, Alexander Kolesnikov, Muhammad Ferjad Naeem, Bernt Schiele, and Federico Tombari. Toward a diffusion-based generalist for dense vision tasks. *arXiv preprint arXiv:2407.00503*, 2024.

[14] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, Dong Chen, and Baining Guo. Instructdiffusion: A generalist modeling interface for vision tasks. In *CVPR*, 2024.

[15] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[17] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *ICCV*, 2023.

[18] Aliasghar Khani, Saeid Asgari Taghanaki, Aditya Sanghi, Ali Mahdavi Amiri, and Ghassan Hamarneh. Slime: Segment like me. In *ICLR*, 2024.

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

[20] Neehar Kondapaneni, Markus Marks, Manuel Knott, Rogério Guimaraes, and Pietro Perona. Text-image alignment for diffusion-based perception. In *CVPR*, 2024.

[21] Hsin-Ying Lee, Hung-Yu Tseng, Hsin-Ying Lee, and Ming-Hsuan Yang. Exploiting diffusion prior for generalizable dense prediction. In *CVPR*, 2024.

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

[23] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. In *NeurIPS*, 2022.

[24] Weifeng Lin, Xinyu Wei, Renrui Zhang, Le Zhuo, Shitian Zhao, Siyuan Huang, Junlin Xie, Peng Gao, and Hongsheng Li. Pixwizard: Versatile image-to-image visual assistant with open-language instructions. In *ICLR*, 2025.

[25] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[26] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[28] 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 *ICLR*, 2024.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[30] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

[31] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In *CVPR*, 2024.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[36] Wouter Van Gansbeke and Bert De Brabandere. A simple latent diffusion approach for panoptic segmentation and mask inpainting. *arXiv preprint arXiv:2401.10227*, 2024.

[37] Chaoyang Wang, Xiangtai Li, Henghui Ding, Lu Qi, Jiangning Zhang, Yunhai Tong, Chen Change Loy, and Shuicheng Yan. Explore in-context segmentation via latent diffusion models. *arXiv preprint arXiv:2403.09616*, 2024.

[38] Chaoyang Wang, Xiangtai Li, Lu Qi, Henghui Ding, Yunhai Tong, and Ming-Hsuan Yang. Semflow: Binding semantic segmentation and image synthesis via rectified flow. In *NeurIPS*, 2024.

[39] Mengyu Wang, Henghui Ding, Jun Hao Liew, Jiajun Liu, Yao Zhao, and Yunchao Wei. Segrefiner: Towards model-agnostic segmentation refinement with discrete diffusion process. In *NeurIPS*, 2023.

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

[41] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

[42] Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? *arXiv preprint arXiv:2403.06090*, 2024.

[43] Yifeng Xu, Zhenliang He, Meina Kan, Shiguang Shan, and Xilin Chen. Jodi: Unification of visual generation and understanding via joint modeling. *arXiv preprint arXiv:2505.19084*, 2025.

[44] Lehan Yang, Jincen Song, Tianlong Wang, Daiqing Qi, Weili Shi, Yuheng Liu, and Sheng Li. Vrmdiff: Text-guided video referring matting generation of diffusion. *arXiv preprint arXiv:2503.10678*, 2025.

[45] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

[46] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025.

[47] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[49] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023.

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Overview. In the supplementary materials, we present more results and details:

- **A.** More details on the method.
- **B.** More empirical studies and visual results.
- **C.** Discussion of the broader impacts, safeguards and limitations.
- **D.** Compute Reporting.

A Implementation Details

Our FlowSeg model is trained with 80K iterations for stage 1 and 40K iterations for stage 2 on the ADE20K dataset, and with 320K iterations for stage 1 and 160K iterations for stage 2 on the COCO-Stuff dataset. The maximum gradient norm is clipped to 1.0. To further validate the effectiveness of our method on a dataset with higher label cardinality, we additionally evaluate it on ADE20K-847. Unless otherwise specified, ADE20K refers to the standard 150-class setting (ADE20K-150).

During training, segmentation masks may include invalid regions, so we apply a binary mask M to filter out these regions in both our method and the flow matching baselines, as defined in Eq. 17. In the main manuscript, we omit it for brevity in line with standard practices in the segmentation field.

$$\mathcal{L}_{res} = \mathbb{E}\left[\left\|(v_\theta(x_t, t, I) - \text{sg}[\tilde{v}_t]) \odot M\right\|^2\right] \tag{17}$$

During inference, the predicted mask is compared against the nearest semantic centroid, and the corresponding category is assigned accordingly.

We summarize the main hyperparameters in Tab. 3 and Tab. 4. The main experiments use the full two-stage schedule, whereas ablations use only the first stage, trained for 20K iterations.

Table 3: Setting in Stage 1.

Hyperparameter	**Value**
Input resolution	256×256
Total batch size	64
Backbone	PixNerd-XL-P16-C2I
Patch size	16
Hidden dimension	1152
Transformer blocks	30
Conditional blocks	26
Input / output channels	3 / 3
Pretraining	PixNerd-XL-P16-C2I checkpoint
Optimizer	AdamW
Learning rate	5×10^{-5}
Adam betas	$(0.9, 0.999)$
Weight decay	0.01
Gradient clipping	1.0

B Empirical Results

B.1 Pixel Space Modeling

We employ end-to-end pixel-space modeling rather than traditional latent-space modeling with VAEs. The VAE framework constructs a distorted latent space for optimization, which diminishes pixel-level distinctions to the extent that they become indistinguishable. This characteristic makes VAEs inherently unsuitable for segmentation tasks. As shown in Fig. 6, VAEs exacerbate this issue: although the predicted colors appear nearly identical, minor differences can lead to assignment to entirely different semantic categories, resulting in the *scattered* effect highlighted in the red box.

Table 4: Setting in Stage 2.

Hyperparameter	**Value**
Input resolution	512×512
Total batch size	16
Backbone	PixNerd-XL-P16-C2I
Patch size	16
Hidden dimension	1152
Transformer blocks	30
Conditional blocks	26
Input / output channels	3 / 3
Pretraining	Stage 1 checkpoint
Optimizer	AdamW
Learning rate	1×10^{-5}
Adam betas	$(0.9, 0.999)$
Weight decay	0.01
Gradient clipping	1.0

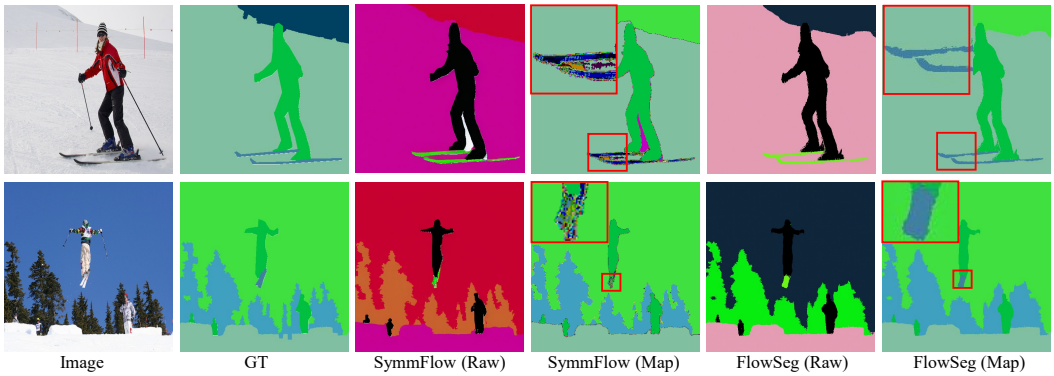


Figure 6: Visualization of the impact of VAE on the final prediction. **GT** denotes the ground truth. The **Raw** prediction is the model's direct, unprocessed output. To obtain the **Map** prediction, we compute the distance between each pixel and the candidate centroids, and assign each pixel to the category of its nearest centroid. In this comparison, SymmFlow uses latent-space modeling with a VAE, whereas our FlowSeg adopts end-to-end pixel-space modeling.

B.2 Convergence Analysis

To better highlight the differences between FlowSeg and vanilla flow matching, we extend the training schedule to 200K iterations with a batch size of 64, corresponding to over 600 epochs, ensuring convergence. As shown in Fig. 7, FlowSeg consistently outperforms vanilla FM throughout the training process, demonstrating both faster convergence and superior performance.

B.3 Scaling to Higher Cardinality

We evaluate our method on ADE20K-847, a challenging semantic segmentation benchmark with 847 categories, substantially more than most commonly used datasets. We follow the two-stage training procedure described in the main manuscript. The results are reported in Tab. 5.

The results demonstrate that FlowSeg remains effective under extreme class cardinality, achieving performance comparable to the specialized discriminative model SimSeg, while substantially outperforming the diffusion-based baseline PromptDiffusion, which struggles in this setting.

B.4 Sensitivity to Temperature

We first evaluate the temperature parameter τ within the softmax assignment. As shown in Tab. 6, the optimization dynamics are sensitive to this scaling. Low temperature yields an overly sharp

Table 5: Comparison on ADE20K-847.

Model	Type	mIoU
SimSeg	Discriminative Specialists	10.43
PromptDiffusion	Diffusion Models	0.99
FlowSeg (Ours)	Diffusion Models	**10.77**

assignment, destabilizing the gradient and severely bottlenecking performance. Conversely, excessive temperatures flatten the distance-aware penalty, diluting the discriminative signal. The sweet spot of τ is around 0.5 to 1, indicating that the reshape field relies on moderate sharpness to deliver gradients that are both semantically informative and numerically stable.

Table 6: **Temperature ablation.** We vary the temperature τ in the distance-to-probability transformation while keeping other settings constant. Moderate sharpness yields the best performance.

Dataset	0.25	0.50	1.00	2.00	5.00
ADE-150	20.17	27.29	**29.75**	23.36	19.73
ADE-847	2.75	**4.59**	4.53	3.15	3.02

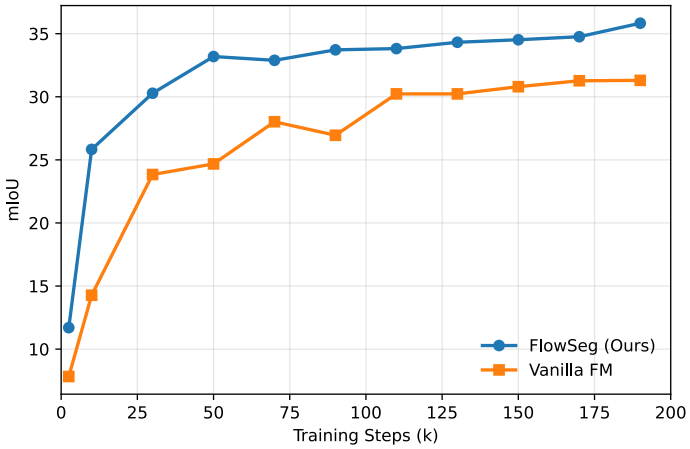


Figure 7: Comparison with vanilla FM on the ADE20K dataset under a long training schedule (200K iterations, batch size 64). The resolution is set to 256×256.

B.5 Centroid Assignment

We conduct an ablation study on centroid assignment. Specifically, we treat the previously ignored category as an additional valid class, resulting in a total of $K+1$ centroids. During inference, although the model outputs raw predictions for all centroids, the nearest label assignment is performed only among the K valid centroids. As shown in Tab. 7, including an additional centroid has a limited impact on performance (Rows 1^{st} and 3^{rd}). Instead, performance is more sensitive to mask filtering (Eq. 17), which is consistent with our conclusions in the main manuscript.

We conduct experiments on the ADE-150 dataset, considering two training recipes: vanilla flow matching and FlowSeg, and two initialization methods: Kronecker-style and PCA-style. The latter incorporates semantic information. Specifically, for each category in the ADE-150 dataset, we used the prompt "a photo of {CLASS NAME}" to extract text embeddings via CLIP ViT-B, then applied PCA to reduce the dimensionality to three and normalized the results to [-1, 1]. The results are shown in Tab. 8.

Table 7: Ablations on centroid assignment. Additional-Centroid indicates assigning a specific centroid for the ignored category.

Additional-Centroid	Mask	mIoU
-	✓	29.75
✓	-	26.80
✓	✓	29.21

Table 8: Effect of centroid initialization on vanilla FM and FlowSeg.

Method	Init.	mIoU	Min. NN Dist.
Vanilla FM	PCA	24.16	0.02
Vanilla FM	Kronecker	21.14	0.15
FlowSeg	PCA	28.42	0.02
FlowSeg	Kronecker	29.75	0.15

B.6 Learning Rate Compensation

We observe that the difference in convergence, particularly the gradient vanishing phenomenon, between our FlowSeg and vanilla flow matching stems from the underlying optimization dynamics and cannot be mitigated simply by increasing the learning rate. As shown in Fig. 8, naively raising the learning rate results in significant performance degradation. This is because an excessively high learning rate causes regions far from the centroid to experience extremely large updates, leading to severe optimization oscillations and preventing convergence to the optimal solution. On the other hand, a learning rate that is too low also results in suboptimal performance due to insufficient optimization.

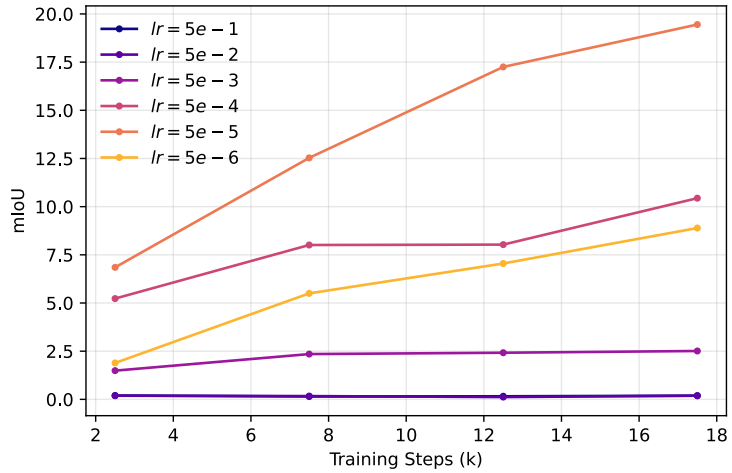


Figure 8: Vanilla FM with different learning rates. $5e-5$ is the default setting.

B.7 Generative Property Analysis

FlowSeg inherits the intrinsic characteristics of generative models. Built upon the ODE framework of flow matching, we employ an Euler solver to evaluate performance across various sampling steps (Tab. 9). On the ADE20K dataset, FlowSeg achieves peak performance at 10 steps; however, efficiency-accuracy trade-offs arise, as too few steps lead to discretization errors, while excessive steps may accumulate approximation errors. As visualized in Fig. 9, the intermediate sampling states exhibit smooth evolution trajectories between the image and mask distributions, confirming that our model preserves the continuous mapping properties of flow-based generation.

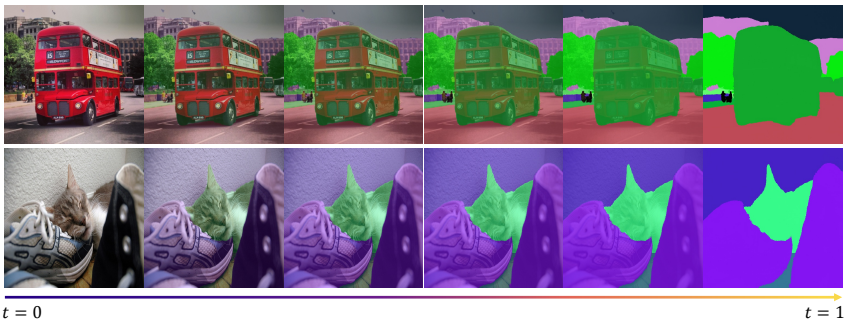


Figure 9: Visualization of intermediate states during sampling.

Table 9: Sampling with different steps on the ADE20K dataset. Euler sampler is adopted.

Step	1	2	5	10	20	30	50	100
mIoU	43.47	45.85	46.82	**47.1**	46.78	46.59	46.13	45.78

C Discussion

C.1 Broader Impacts

This work explores and analyzes the limitations of traditional diffusion-based segmentation methods and proposes a novel vector field reshaping approach that significantly improves the model's discriminative ability while preserving generation capability. We believe this method holds promise for further bridging the gap between generation and understanding, and ultimately promoting a unified solution.

C.2 Safeguards

Given that the primary task of this work is perception and that the generated data are mostly segmentation masks, it does not produce harmful content compared to standard generative models. Furthermore, the datasets used are publicly available and have been extensively validated by the community.

C.3 Limitations

Given that this work primarily explores the feasibility of the concept, we do not focus on architectural modifications or the scaling laws that have been proven effective in engineering, and some segmentation failures exist. However, we believe this method has not yet reached its limits; a better architectural design could achieve even more powerful results, which we leave for future work.

D Compute Reporting

All experiments are conducted on a single-node cluster equipped with Intel(R) Xeon(R) Gold 6248 CPUs @ 2.50GHz, 8 NVIDIA A100 GPUs, and 1.5 TB of system memory.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope

   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: in the appendix

   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: [N/A]

Justification: the paper does not include theoretical results

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: we provide implementation details

   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: project resources are available on our GitHub page

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: we provide implementation details

   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 evaluation focuses on existing benchmark comparisons without error bars

   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: we provide them in the implementation details

   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: the research conducted in the paper conform, in every respect, with 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: in the appendix

    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: [Yes]

    Justification: in the appendix

    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: creators are properly credited

    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: [N/A]

    Justification: the paper does not release new assets

    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

    Guidelines:

    - 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: the core method development in this research does not involve LLMs as any important, original, or non-standard components

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.