SPATIALLY-GROUNDED FLOW MATCHING: STRUCTURED SOURCE DISTRIBUTIONS FOR IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current flow matching models learn to transport the source i.i.d. Gaussian noise into the target distribution of natural images, yet this source distribution carries no notion of spatial structure. Images however are fundamentally local since nearby pixels are strongly correlated. We hypothesize that sampling the noise independently implicitly encourages models to exploit less noisy neighbors as context during training, partially bypassing the need to properly learn the true local structure of images. The source distribution, in other words, works against the inductive bias of the image domain. To ameliorate this design discrepancy, we propose StructFlow which encodes spatial locality directly into the source by having the pixels within a small region share a common noise component. This structured source produces transport paths that are geometrically aligned with image regions, enabling properties that generic flow matching struggles to provide: fine-grained local editing that naturally respects boundaries, robust structure preservation, and smooth semantic interpolation between images. We show that these benefits also extend to large pre-trained models, demonstrating that StructFlow can even be incorporated through a lightweight post-training phase. Comprehensive experiments on multiple datasets, in unconditional, class and text-conditioned regimes, using different diffusion transformer architectures confirm that StructFlow not only offers competitive image generation quality, but also significantly improves localized controllable re-synthesis.

1 INTRODUCTION

Recent advances in image generative modeling, particularly diffusion models and flow matching (Ho et al., 2020; Rombach et al., 2022; Nichol & Dhariwal, 2021; Ho & Salimans, 2022; Labs, 2024; Esser et al., 2024; Lipman et al., 2022; Wu et al., 2025), have led to remarkable improvements in visual quality and scalability, establishing state-of-the-art across a wide range of tasks (Ruiz et al., 2023; Brooks et al., 2023; Sohn et al., 2023; Wang et al., 2024; Zhang et al., 2023). In flow matching, generation is formulated as learning a continuous transport from a simple source distribution (typically an isotropic Gaussian) to the data distribution via a learned velocity field. Despite its success, this standard choice of source distribution is largely inherited from diffusion models and remains under-explored. A key limitation of this formulation is a fundamental mismatch between the source distribution and that of natural images. While images exhibit strong spatial locality, where nearby pixels are highly correlated and form coherent regions, the commonly used i.i.d. Gaussian prior assumes complete independence across spatial locations. As a result, the model is implicitly required to reconstruct this locality through its learned dynamics, rather than benefiting from it as an inductive bias. This places an unnecessary burden which we hypothesize would hinder the training efficiency as well as controllability of the final model. Several recent works have explored alternatives to the standard Gaussian source distribution, including non-Gaussian priors (Nachmani et al., 2021; Xu et al., 2022), mixture-based formulations (Chen et al., 2025; Jia et al., 2024), and learned source distributions (Kim et al., 2026; Lee et al., 2025). While these approaches improve aspects such as convergence and expressivity, they largely remain agnostic to the spatial structure inherent in the natural images. In contrast, structured noise has proven highly effective in other domains: for instance, recent video diffusion models introduce temporally correlated noise to enforce consistency across frames (Ge et al., 2023; Chang et al., 2024), leading to significantly improved coherence. This raises a natural question: *can similar structured priors be leveraged in the spatial domain for image generation?*

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

In this work, we introduce *StructFlow*, a flow matching framework with spatially-grounded source distributions that explicitly encodes locality. Our key idea is to construct a structured prior in which pixels within a small region share a common noise component, inducing a bias that aligns with local structure in natural images. This simple modification leads to transport paths that are geometrically consistent with image regions, enabling emergent capabilities such as fine-grained local editing, improved structure preservation, and controllable coherence during generation. However, a naive implementation of such a structured source distribution introduces optimization challenges, mainly due to increased correlations in the noise, which severely destabilizes the training. To facilitate the learning process, we propose (i) a cosine-induced spatial anchoring which reduces stochastic variability while providing consistent spatial cues, (ii) a progressive coherence annealing schedule that gradually introduces structure during training, and (iii) a mixed-coherence training strategy that enables a single model to operate across a spectrum that trades generation quality for localized controllable re-synthesis capabilities. We validate StructFlow across multiple problem settings, including class and text-conditioned image generation, demonstrating that our approach achieves competitive image quality while significantly improving the granularity of control when it comes to local edits (*i.e.* re-synthesis). Notably, our method can also be incorporated into pre-trained models through a lightweight post-training, making it applicable to existing foundation generative models.

In summary, our contributions are: (1) We identify the mismatch between i.i.d. source distributions and spatial locality in images as a key limitation of standard flow matching and propose *StructFlow*, a spatially structured source distribution that encodes locality directly into the generative process. (2) We introduce a set of training strategies that stabilize optimization under structured noise while preserving its benefits. (3) We demonstrate improved controllability, structure preservation, and editing capabilities without sacrificing generation quality across multiple setups and benchmarks.

2 Background

Our work builds on flow matching (Lipman et al., 2022; Liu et al., 2022), which learns a time-dependent vector field to transport samples from a source distribution to the data distribution via an ODE (see Appendix A.1 for details). A key design choice in flow matching is the source distribution, which is typically taken to be a standard isotropic Gaussian, largely inherited from diffusion models rather than optimized for the task. A growing body of work challenges this assumption. For instance, Cold Diffusion (Bansal et al., 2023) shows that arbitrary image degradations can define valid generative processes; Non-Gaussian Diffusion (Nachmani et al., 2021) and Poisson Flow (Xu et al., 2022) explore alternative priors such as Gamma distributions and hemisphere-uniform sampling; and Gaussian Mixture Flow Matching (Chen et al., 2025) and Structured Diffusion with MoG (Jia et al., 2024) employ mixture-based priors for richer distributional coverage. More directly related to our setting, several works investigate the design of source distributions in flow matching (Kim et al., 2026; Lee et al., 2025; Zwick et al., 2025), demonstrating that carefully chosen or learned priors can significantly improve convergence speed and generation quality. Whitened Score Diffusion (Alido et al., 2025) further highlights the benefits of anisotropic Gaussian priors by introducing spectral inductive biases tailored to imaging tasks. Additional approaches construct non-standard priors via normalizing flows (Zand et al., 2023) or diffusion bridges (Zhou et al., 2023), or introduce spatially varying noise through time-dependent scheduling on pre-trained models (Yu et al., 2023). Beyond images, recent work has shown that introducing structured correlations into the noise prior, rather than treating pixels independently, can improve coherence in video generation. For example, PYoCo (Ge et al., 2023) proposes a mixed noise prior for video diffusion in which frames share a common noise component, enabling strong temporal consistency. Similarly, (Chang et al., 2024) introduces a principled method to warp noise across video frames using optical flow, preserving noise statistics while enforcing temporal alignment.

Our work complements and extends these directions by introducing a spatially structured source distribution that explicitly captures the locality inherent in natural images. We show that this inductive bias can be incorporated in a particularly simple way through the source distribution itself, rather than relying solely on the model to learn such structure from data under an unstructured Gaussian prior. Concretely, we construct priors in which pixels within the same spatial neighborhood share correlated noise, thereby promoting spatial coherence throughout the flow matching process. As a result, the model naturally learns to preserve structural consistency during generation, while enabling emergent capabilities such as localized editing for free.

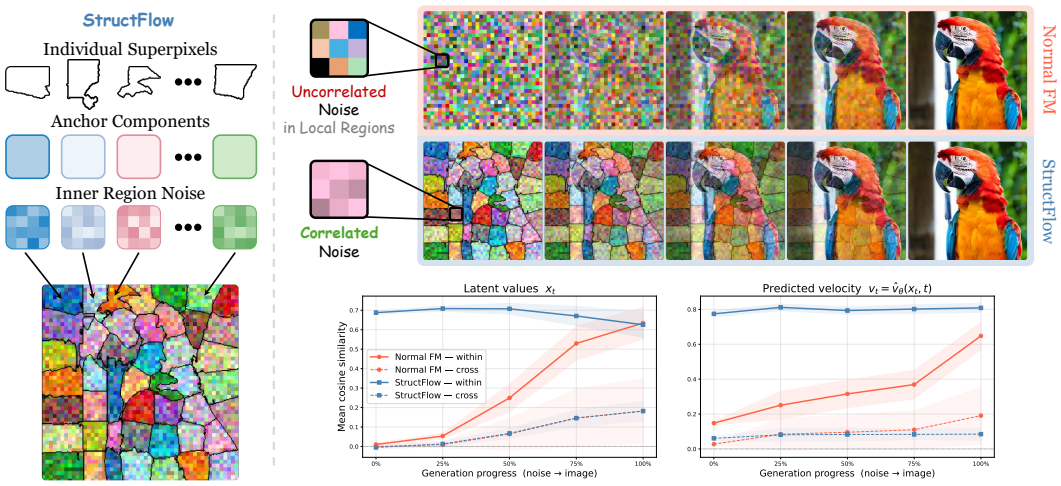


Figure 1: **StructFlow Noise Construction and locality analysis.** StructFlow introduces locality in the source distribution instead of i.i.d gaussian sampling in normal flow matching. Locality emerges implicitly in standard flow matching, while StructFlow enforces it directly in the source distribution.

3 METHOD

3.1 FLOW MATCHING AND LOCALITY BIAS

As a starting point, we briefly review the standard flow matching (Lipman et al., 2022) framework, in particular rectified flow (Liu et al., 2022), upon which our method is built. Flow matching constructs a continuous-time transport map between a source distribution p_0 and the data distribution p_1. In practice, for image generation, the source distribution is typically chosen as an isotropic Gaussian, $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$, such that each pixel (or latent dimension) is sampled independently. Given a data sample $\mathbf{x}_1$ and a source sample $\boldsymbol{\epsilon} \sim p_0$, rectified flow considers the linear interpolation path

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

and learns a neural velocity field $v_\theta(\mathbf{x}_t, t)$ to predict the corresponding constant velocity, $\mathbf{x}_1 - \boldsymbol{\epsilon}$.

Natural images exhibit a strong *locality inductive bias*: nearby pixels are highly correlated and form coherent spatial regions. In contrast, the standard choice of source distribution, $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$, is fully factorized, producing spatially independent noise. As a result, the model must reconstruct this locality entirely through the learned velocity field, forcing the model to reconstruct spatial structure through the learned velocity field, rather than inheriting it from the source distribution.

To study this effect, we analyze a DiT-based flow matching model. We sample generated trajectories and record both the latent state $\mathbf{x}_t^i$ and predicted velocity $v_\theta(\mathbf{x}_t^i, t)$ at multiple timesteps $t \in \{0, 0.25, 0.5, 0.75, 1\}$. For each generated sample $\mathbf{x}_1^i$, we compute a SLIC (Achanta et al., 2012) superpixel segmentation $\mathbf{M}^i$ and measure feature similarity within and across segments in both $\mathbf{x}_t^i$ and $v_\theta(\mathbf{x}_t^i, t)$. Let $\mathbf{f} \in \{\mathbf{x}_t,\ v_\theta(\mathbf{x}_t, t)\} \in \mathbb{R}^{N_p \times C}$ denote the pixel features (flattened spatial dimensions) and $\hat{\mathbf{f}}_p = \mathbf{f}_p / \|\mathbf{f}_p\|$ the ℓ_2-normalised features. Given a segmentation $\mathbf{M}^i$ with segments $\{\mathcal{S}_k\}_{k=1}^K$ ($\mathcal{S}_k$ denotes the set of pixels in k-th segment), we define the average within-segment and cross-segment similarities as $\rho_{\text{in}}(\mathbf{f}, \mathbf{M}) = \mathbb{E}_k\, \mathbb{E}_{p,q \in \mathcal{S}_k} \left[\hat{\mathbf{f}}_p^\top \hat{\mathbf{f}}_q\right]$ and $\rho_{\text{out}}(\mathbf{f}, \mathbf{M}) = \mathbb{E}_k\, \mathbb{E}_{p \in \mathcal{S}_k,\, q \notin \mathcal{S}_k} \left[\hat{\mathbf{f}}_p^\top \hat{\mathbf{f}}_q\right]$ where:

$$\rho_{\text{in}}(\mathbf{f}, \mathbf{M}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{\binom{|\mathcal{S}_k|}{2}} \sum_{\substack{p,q \in \mathcal{S}_k \\ p<q}} \hat{\mathbf{f}}_p^\top \hat{\mathbf{f}}_q \qquad \rho_{\text{out}}(\mathbf{f}, \mathbf{M}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{S}_k||\bar{\mathcal{S}}_k|} \sum_{p \in \mathcal{S}_k,\, q \in \bar{\mathcal{S}}_k} \hat{\mathbf{f}}_p^\top \hat{\mathbf{f}}_q \tag{2}$$

Findings. As shown in Fig. 1 , the latent state $\mathbf{x}_t$ exhibits almost no spatial structure at initialization ($\rho_{\text{in}}(x_t, M) \approx 0$ at t=0), and gradually acquires locality over time. In contrast, the velocity field already encodes strong spatial correlations from the earliest steps: even at t=0, we observe a clear gap $\rho_{\text{in}} - \rho_{\text{out}}$, which increases steadily during generation. This indicates that the model learns to impose spatial structure through its velocity field, compensating for the lack of structure in the source distribution. In other words, part of the model capacity is spent reconstructing locality that could instead be built into p_0. This observation motivates our central question: *can we design a source distribution p_0 that already respects the locality structure of images, relieving the model of this burden and unlocking emergent capabilities?*

Algorithm 1 StructFlow Hierarchical Noise Sampling

Input: Mask $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{H \times W}$ with segments $\{\mathcal{S}_k\}_{k=1}^{K}$, latent spatial size (C, H, W), secondary noise scale $\lambda_\varsigma \geq 0$, cosine grid $g \in [-1, 1]^{H \times W}$
Output: Structured noise $\boldsymbol{\epsilon} \in \mathbb{R}^{C \times H \times W}$

*# **1st Level**: Sample one shared noise vector per segment*
for $k = 1, \ldots, K$ **do**
  $\mathbf{z}_k \leftarrow \begin{cases} |\mathcal{S}_k|^{-1} \sum_{(i,j) \in \mathcal{S}_k} g(i,j), & \text{if } \texttt{CISA}, \\ \mathcal{N}(\mathbf{0}, \mathbf{I}_C), & \text{otherwise.} \end{cases}$
end for
*# **2nd Level**: Sample per-pixel noise per shared segment*
for each spatial location (i, j) **do**
  $\boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(\mathbf{z}_{\mathbf{M}(i,j)}, \lambda_\varsigma^2 \mathbf{I}_C)$
end for
return $\boldsymbol{\epsilon} / \sqrt{1 + \lambda_\varsigma^2}$ ▷ Rescale to unit variance

Algorithm 2 StructFlow Training

Input: Dataset $\mathcal{D}$, Model v_θ, # of Superpixels K
for each training step s **do**
  Sample $\mathbf{x}_1 \sim \mathcal{D}$
  $\mathbf{M} = \texttt{SLIC}(\mathbf{x}_1, K)$
  if `Progressive_Coherence_Annealing` **then**
    Update $\lambda_\varsigma \leftarrow \text{schdeuler}_{\lambda_\varsigma}(s, \lambda_{\min}, \lambda_{\max})$
  end if
  if `Mixed_Coherence` & $\text{schdeuler}_{\lambda_\varsigma}$.finish **then**
    Sample $\lambda_\varsigma \sim \mathcal{U}[\lambda_{\min} - \delta, \lambda_{\min} + \delta]$
  end if
  Sample $\boldsymbol{\epsilon}$ using Alg. 1 and $t \sim \mathcal{U}(0, 1)$
  $\mathbf{x}_t \leftarrow (1 - t)\boldsymbol{\epsilon} + t\mathbf{x}_1$
  $\hat{\mathbf{v}}_\theta \leftarrow v_\theta(\mathbf{x}_t, t)$
  $\mathcal{L} \leftarrow \|\hat{\mathbf{v}}_\theta - (\mathbf{x}_1 - \boldsymbol{\epsilon})\|_2^2$
  Update θ with $\nabla_\theta \mathcal{L}$
end for

3.2 Structured Hierarchical Noise Sampling

We introduce *StructFlow*, which replaces the isotropic Gaussian source with a *structured* distribution that encodes spatial locality. The key idea is to induce correlations among pixels within the same region, while preserving sufficient stochasticity to avoid regional collapse to a single latent value.

Let $\mathbf{M} \in \mathbb{Z}_{\geq 0}^{H \times W}$ denote a superpixel decomposition with K regions, obtained via SLIC oversegmentation (Achanta et al., 2012) (or a proxy at inference; see Section 4.5). We operate in the VAE latent space of shape (C, H, W) and define a hierarchical Gaussian prior in which pixels within the same segment share a common latent anchor. Concretely, for each segment $k \in \{1, \ldots, K\}$, we sample an anchor vector $\mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_C)$, and for each spatial location (i, j) belonging to segment $k = \mathbf{M}(i, j)$, we sample $\boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}\big(\mathbf{z}_k,\ \lambda_\varsigma^2 \mathbf{I}_C\big)$, followed by a global rescaling $\boldsymbol{\epsilon} \leftarrow \frac{\boldsymbol{\epsilon}}{\sqrt{1+\lambda_\varsigma^2}}$ to ensure unit marginal variance. Full sampling procedure is illustrated in Alg. 1 and in Fig 1. This construction induces a simple structure: pixels within the same local region have correlation

$$\mathrm{Corr}(\epsilon_{i,j},\ \epsilon_{i',j'}) = \frac{1}{1 + \lambda_\varsigma^2} \quad \text{for } \mathbf{M}(i, j) = \mathbf{M}(i', j'), \tag{3}$$

while pixels from different segments remain uncorrelated. The parameter λ_ς therefore directly controls the strength of locality, smoothly interpolating between fully correlated segments ($\lambda_\varsigma \to 0$) and standard i.i.d. Gaussian noise ($\lambda_\varsigma \to \infty$) (See Fig 23 in Appendix).

Aside from the structured source $\boldsymbol{\epsilon}$, we make no modifications to the flow matching training objective or inference procedure. The model is trained to predict $\hat{v}_\theta(\mathbf{x}_t, t)$ with the standard MSE loss, and DDIM/Euler integration is used at inference, with the only change being that the initial noise is drawn from our structured distribution rather than $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Encoding locality in the source distribution, as reflected in the analysis in Fig. 1, makes local regions more coherent and better separated from the rest of the image throughout generation. This gives rise to several downstream properties *out of the box*, including *local region editing* (Sec. 4.3.1), *structure-preserving generation* (Sec. 4.3.2), and the *early emergence of informative features* (Sec. 4.3.3).

3.3 StructFlow Training Strategies

Naïvely training a DiT on the structured source distribution described above can lead to optimization instability: the induced correlations increase gradient variance, making it difficult for the model to learn a consistent velocity field from highly correlated noise. We identify three complementary strategies that resolve this instability while preserving, and enhancing, the locality capabilities.

3.3.1 Cosine-Induced Spatial Anchoring

In the basic formulation (Alg. 1), each segment anchor $\mathbf{z}_k$ is sampled independently from a Gaussian distribution. Although this creates within-segment coherence, it can also assign very different anchors to adjacent segments within the same training sample, introducing abrupt high-variance

discontinuities in the structured source that make the target velocity field harder to learn. We find that replacing this random sampling with a reduced-stochasticity spatial structure substantially stabilizes training. Concretely, we define a diagonal cosine grid over the spatial lattice:

$$g(i,j) = 2 \cdot \frac{1 - \cos\left(\pi \cdot \frac{i+j}{2(H-1)}\right)}{2} - 1 \ \in [-1,1], \quad i \in [0, H-1],\ j \in [0, W-1]. \tag{4}$$

Each segment k is assigned the mean grid value over its pixels, $\bar{g}_k = |\mathcal{S}_k|^{-1} \sum_{(i,j)\in\mathcal{S}_k} g(i,j)$, and the first-level anchor is replaced by the scalar broadcast $\mathbf{z}_k \leftarrow \bar{g}_k \cdot \mathbf{1}_C$ (See Fig 22 in Appendix). This Cosine-Induced Spatial Anchoring (CISA) provides two benefits: (i) the first-level structure is deterministic given the mask, removing one source of stochastic variation from the training signal; (ii) the smooth diagonal gradient gives the model a consistent spatial cue which it can leverage throughout training. Overall, this yields a more well-conditioned learning problem, leading to improved optimization stability as shown in Sec 4.2 and Table 1.

3.3.2 Progressive Coherence Annealing

During our experiments, we realized that the noise scale λ_ς governs a fundamental *quality–editability trade-off*: small λ_ς yields tightly coherent segments that enable precise local editing, but makes the source distribution far from Gaussian, destabilizing training. Large λ_ς approaches the i.i.d. baseline, easing optimisation at the cost of weaker locality guarantees.

We resolve this tension with a *coherence annealing* schedule: λ_ς is initialised to a large value $\lambda_{\max}$ and linearly decreased to a target $\lambda_{\min}$ over an interval of training steps $[s_0, s_1]$:

$$\lambda_\varsigma(s) = \lambda_{\max} - (\lambda_{\max} - \lambda_{\min}) \cdot \frac{\min(s, s_1) - s_0}{s_1 - s_0}, \quad s \geq s_0. \tag{5}$$

This curriculum first trains the model on a near-Gaussian source (easy), then progressively tightens the locality constraint as the model matures. Coherence annealing enables a particularly practical workflow: a standard flow matching model can be *fine-tuned* into a StructFlow model by applying the annealing schedule starting from the pretrained checkpoint. Because the initial $\lambda_\varsigma = \lambda_{\max} \to \infty$ is close to the training distribution of the base model, fine-tuning is stable; the model gradually adapts to the structured source without catastrophic forgetting. We demonstrate this in Sec. 4.4.

3.3.3 Mixed-Coherence Training

Even with annealing, committing to a single final λ_ς at test time forces a binary choice between generation quality (high λ_ς) and locality control (low λ_ς). We instead propose *mixed-coherence training*: As the annealing schedule approaches $\lambda_{\min}$, at each training iteration, λ_ς is sampled uniformly from a range $[\lambda_{\min} - \delta, \lambda_{\min} + \delta]$. This provides two advantages. First, training across a distribution of coherence levels acts as a regulariser, improving generation quality even at the hard end ($\lambda_\varsigma = \lambda_{\min} - \delta$) compared to training at that value alone. Second, a single trained model can serve the full spectrum of use cases at inference: low λ_ς for fine-grained local editing and structure preservation, high λ_ς for maximum image quality with mild locality bias. Users can thus navigate the quality–editability pareto *without retraining*. Algorithms 1 and 2 summarize our training schema.

4 Experiments

In this section, we conduct comprehensive quantitative and qualitative evaluations of StructFlow, demonstrating strong performance across a wide range of metrics while revealing compelling emerging capabilities that arise naturally from our approach.

4.1 Implementation details

We use DiT-XL (Peebles & Xie, 2023) as the base architecture in all experiments, building a flow matching implementation similar to SiT (Ma et al., 2024). Unless otherwise stated, all models are trained from scratch. We use a global batch size of 128 for class-conditional ImageNet and 64 for unconditional FFHQ. For class-conditional training, we apply class dropout with probability 0.1 to enable classifier-free guidance (CFG) at inference, where we use a fixed guidance scale of 4.0 across all evaluations. We train on FFHQ for 130k iterations and on ImageNet for approximately 3M iterations unless mentioned otherwise. Additional implementation details are provided in Appendix B.1.



(a) Unconditional (b) Class-Conditional (c) Text-to-Image

Figure 2: **Qualitative Examples.** StructFlow produces high-quality and diverse samples across unconditional, class-conditional, and text-to-image generation settings.

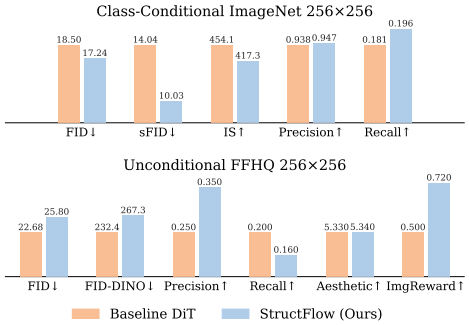
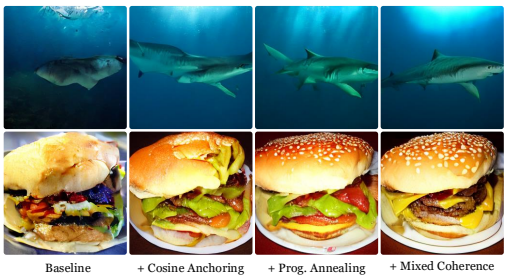


Figure 3: **Comparison with baseline.** Evaluated across a diverse set of metrics, StructFlow performs on par with standard flow matching and outperforms it in some cases.

Table 1: Ablation of training techniques

Class-Conditional ImageNet 256×256					
Training Setting (1M steps)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
Baseline	18.12	14.51	120.26	0.67	**0.23**
+ Cosine Anchoring	11.52	11.27	241.95	0.80	0.19
+ Progressive Annealing	10.97	10.43	330.00	0.85	0.18
+ Mixed Coherence	**10.48**	**10.24**	**346.51**	**0.86**	0.19



4.2 Evaluation

We first evaluate how StructFlow compares to standard flow matching (FM) when training DiT models. To ensure a fair comparison, we train models under identical settings (e.g., training iterations, batch size; see Appendix B.1 for full details) on both unconditional and class-conditional datasets.

Figure 3 presents the quantitative comparison across a diverse set of metrics, including FID (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), DINO-based (Oquab et al., 2023) FID, sFID (Nash et al., 2021), Precision/Recall (Kynkäänniemi et al., 2019), Aesthetic score (Schuhmann & Beaumont, 2022), and ImageReward (Xu et al., 2023). These metrics collectively capture both fidelity and diversity, enabling a comprehensive evaluation. As shown in Figure 3, StructFlow performs on par with standard FM, and even surpasses it on several metrics. Specifically, on class-conditional ImageNet, StructFlow matches or outperforms standard FM on FID, sFID, Precision, and Recall, while falling slightly behind on Inception Score. For unconditional FFHQ, StructFlow similarly performs on par with standard FM on FID and Aesthetic score, while outperforming it on several metrics, including Precision and ImageReward. This demonstrates that StructFlow preserves the strong generative quality of FM while enabling additional structured capabilities. Figure 2 further illustrates qualitative samples from models trained with StructFlow across unconditional, class-conditional, and text-conditioned (see Sec. 4.4) settings. The generated images exhibit high visual quality and diversity, confirming that StructFlow does not compromise sample realism while unlocking emerging capabilities. Refer to Appendix B.2 for more details.

We conduct an ablation study to analyze the contribution of each training strategy proposed in Sec. 3.3. Starting from the naive hierarchical noise sampling (Sec 3.2), we incrementally introduce each component while keeping all other training settings fixed. All models are trained for 1M iterations to ensure a consistent comparison. As shown in Table 1, accompanied with a few qualitative samples, each added component contributed to training stability and overall sample quality, with the full model achieving the best performance. These results highlight the importance of our design choices and demonstrate that the proposed techniques lead to more reliable training and better generative performance.

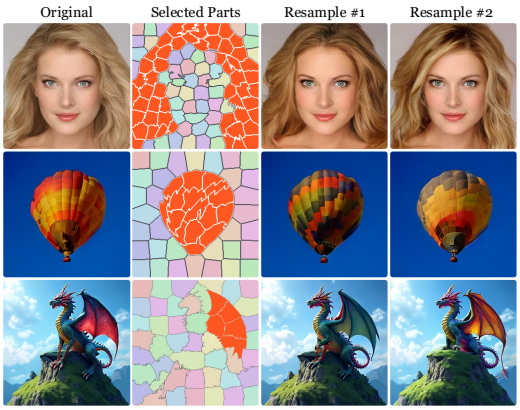


Figure 4: StructFlow allows fine-grained and local edits by resampling the noise in the selected local regions while keeping it fixed elsewhere.

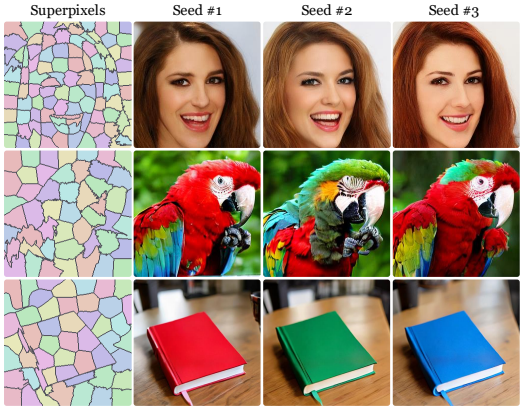


Figure 5: StructFlow allows generation of diverse samples using the same grounding superpixel masks while preserving the overall structure.

4.3 Applications

In this section, we highlight the emerging capabilities and interesting behaviors that StructFlow enables out of the box. These experiments are not intended to position StructFlow as a specialized method for image editing, layout control, or representation learning. Instead, they illustrate properties that arise naturally from introducing spatial structure into the source distribution.

4.3.1 Fine-Grained Image Editing and Part Resampling

A key property of StructFlow is that all pixels within the same segment k share a common noise component, making different segments more clearly disentangled from one another. This structured decomposition enables targeted manipulation at the segment level. In particular, we can selectively resample the secondary noise for a subset of segments $\mathcal{S} = \{k_1, \ldots, k_s\}$, which can be specified either by the user or automatically (e.g., via a segmentation model). By resampling only these segments while keeping the rest fixed, StructFlow generates diverse variations within the selected regions while preserving the remaining content. Unlike SDEdit-style masked editing (Meng et al., 2021), which preserves unmasked regions by explicitly replacing them at each denoising step, StructFlow achieves localized resampling naturally through its structured source distribution without requiring explicit masking operations during denoising.

Figure 4 illustrates this behavior. The first column shows the initial generations, the second highlights the selected regions, and the third and fourth show the outcome if just those selected regions are resampled differently. This enables precise local edits while preserving other regions. For real images, we first invert the image into latent space using standard techniques (e.g., DDIM (Song et al., 2020) or RF inversion (Rout et al., 2025)), then apply segment-wise resampling in the same way. Additional details are in Appendix C.1.

4.3.2 Structure Preserving Generation

StructFlow also enables explicit control over the spatial structure of generated images through its reliance on segmentation masks. In particular, the model naturally adheres to the provided segment boundaries, resulting in generations that respect the underlying structure. This property allows users to design or modify segmentation masks to guide the global layout of the image. As a result, multiple samples generated from the same mask share a consistent structural composition, while still exhibiting diversity in appearance and fine details. Figure 5 demonstrates this behavior. While all generated samples follow the same overall structure, they vary in texture, color, and local details, highlighting StructFlow 's ability to preserve structure without sacrificing diversity.

Figure 6 shows qualitative examples of StructFlow trained on ImageNet, where the class label is swapped at different early inference steps. The left column presents the original samples using the source class, while the subsequent ones show results when switching to the target class at progressively earlier stages. Beyond the smooth semantic transition, StructFlow preserves the underlying spatial



Figure 6: StructFlow's structure preserving capabilities allows interesting class interpolation during the early denoising steps of the generation while preserving structure.

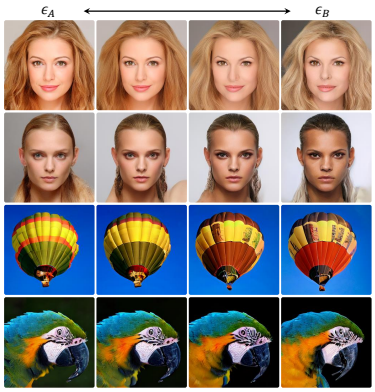


Figure 7: Interpolating initial generation noise ($e_{A,B}$) in StructFlow yields smooth image transitions.



Figure 8: Qualitative comparison of structure preserving methods (best zoomed in).

Table 2: Quantitative comparison of structure-preserving methods.

	Structure		Diversity/Alignment/Quality		
Method	Self-Sim↓	Depth↓	LPIPS↑	CLIP↑	IQA↑
Base SANA	0.222	0.208	0.556	27.97	0.736
ControlNet-HED	0.149	0.122	0.331	26.91	**0.734**
ControlNet-Canny	**0.114**	**0.065**	0.274	27.63	0.708
T2I-Adapter-Canny	0.183	0.153	**0.508**	28.16	**0.734**
SDEdit	0.119	0.076	0.296	27.69	0.719
StructFlow (Ours)	0.181	0.146	0.395	**28.25**	**0.734**

Table 3: Quantitative comparison of local editing methods.

	Structure	Background		Diversity/Quality	
Method	Dist.↓	PSNR↑	LPIPS↓	LPIPS↑	IQA↑
SDEdit	0.048	16.95	0.189	0.150	0.741
Blended Latent Diff.	0.030	27.58	0.017	**0.160**	0.728
SD-1.5 Inpainting	0.027	28.07	0.046	0.153	0.738
SDXL Inpainting	0.016	31.39	0.014	0.136	0.738
InstructPix2Pix	0.015	26.46	0.067	0.044	0.736
FLUX.1-Kontext	0.034	21.19	0.110	0.087	**0.750**
StructFlow	0.042	17.25	0.111	0.148	0.731
StructFlow + Blending	**0.013**	**38.71**	**0.003**	0.103	0.739

structure across classes, enabling consistent and meaningful class-level interpolations. Additional examples are provided in Appendix C.2.

Comparison with other baselines. Furthermore, while our goal is not to achieve state-of-the-art performance in structure preservation or part resampling, these capabilities emerge naturally from our method without specialized training. Nevertheless, we compare against dedicated baselines in Tables 2 and 3 and show that our method achieves competitive performance. Figure 8 presents qualitative comparisons on the structure-preserving task against methods specifically trained for this purpose, where our method performs on par with, and in some cases better than, these specialized approaches. Additional details on the baselines and evaluation metrics, together with further qualitative and quantitative results, are provided in the Appendix G.

4.3.3 Early Emergence of Informative Features

We analyze the intermediate representations of DiT models during the denoising process by recording features at selected layers and timesteps across multiple generations. We then apply PCA to these features and visualize the top three components as RGB images. Interestingly, StructFlow-trained models exhibit highly informative and spatially structured features even at very early timesteps. In particular, coarse object boundaries and layout emerge much earlier compared to models trained with standard flow matching. Figure 9 illustrates this behavior. Refer to Appendix C.3 for more details.

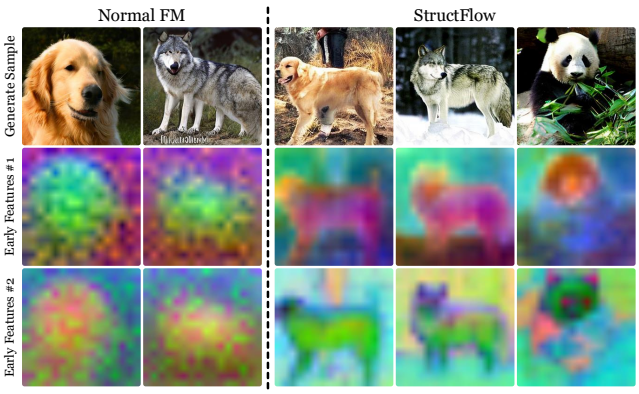


Figure 9: StructFlow's internal features in early denoising inference steps show emergent informative features that can be used for downstream application such as segmentations.

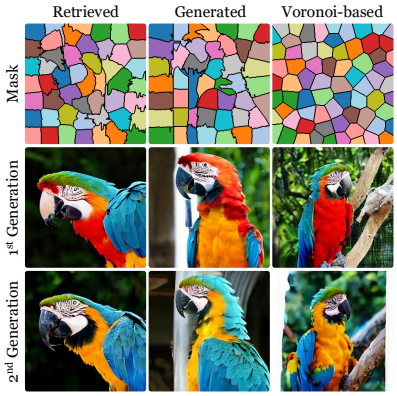


Figure 10: We show that StructFlow produces high-quality results across different mask generation methods.

4.3.4 Latent Interpolation

We study the structure of the StructFlow source distribution by sampling two noise instances and linearly interpolating between them, followed by generation from the interpolated latents. As shown in Figure 7, this results in smooth and coherent transitions in the generated images. This behavior indicates that the latent space induced by StructFlow is well-structured and semantically meaningful, enabling consistent interpolation between samples.

4.4 Text-to-Image StructFlow

As discussed in Sec. 3.3.2, Progressive Coherence Annealing enables post-training of models originally trained with the standard FM objective. Leveraging this property, we post-train SANA text-to-image model (Xie et al., 2024) within the StructFlow framework. Qualitative results in Figure 2 (c) demonstrate high-fidelity image generation, while Figures 5 and 4 highlight key capabilities such as accurate structure preservation and fine-grained editing. Refer to Appendix D for more details.

4.5 Mask Generation

StructFlow relies on segmentation masks during training, readily obtained using superpixel methods such as SLIC (Achanta et al., 2012). At inference, however, such masks are unavailable. We outline several simple strategies to address this, with further details in Appendix E.

First, we consider a retrieval-based approach, where masks are sampled from a pool of precomputed masks. Given that StructFlow can generate diverse outputs from a fixed mask, this already enables diverse image generation. Second, one can train a lightweight generative model to produce masks. These generated masks can then be used to construct the input noise. While effective, this approach introduces an additional model. To avoid this, we also propose training-free alternatives. Superpixel-like structures can be approximated by sampling seed points with a minimum-distance constraint (e.g., Poisson sampling), followed by assigning each pixel to its nearest seed to form a Voronoi tessellation. We further explore simple variants of such constructions and provide a comprehensive ablation study of training-free mask generation strategies in Appendix E. Figure 10 visualizes these approaches, showing that all methods produce reasonable masks and lead to plausible generations, despite the model being trained only with SLIC masks. Refer to Appendix E for more details.

5 Conclusion

We introduced StructFlow, a spatially grounded flow matching framework that replaces the standard i.i.d. Gaussian with a structured source distribution aligned with image locality. By correlating noise within local regions, StructFlow builds spatial coherence directly into the generative process, reducing the burden on the learned velocity field and enabling new forms of controllability. Since naively training with such structured sources can be unstable, we proposed practical training strategies that stabilize optimization and preserve generation quality. Across unconditional, class-conditional, and text-to-image settings, StructFlow demonstrate strong generation quality while enabling fine-grained local editing, structure-preserving generation, smooth interpolation, and more informative early representations. These results suggest that source distribution design is a powerful and underexplored axis for improving both the controllability and interpretability of generative models.

AI USE STATEMENT

We used generative AI tools to improve the readability and presentation of the manuscript and to assist with the implementation of baseline methods used for comparison. All AI-assisted implementations and outputs were manually reviewed and verified by the authors for correctness. Generative AI was not used to develop the proposed method, formulate the research hypotheses, or interpret the experimental results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

Jeffrey Alido, Tongyu Li, Yu Sun, and Lei Tian. Whitened score diffusion: A structured prior for imaging inverse problems. *arXiv preprint arXiv:2505.10311*, 2025.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42(4):1–11, 2023.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36:41259–41282, 2023.

Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 Sketches*, SIGGRAPH '07, pp. 22–es, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781450347266. doi: 10.1145/1278780.1278807. URL https://doi.org/10.1145/1278780.1278807.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pzElnMrgSD.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. *arXiv preprint arXiv:2504.05304*, 2025.

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.

Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. *arXiv preprint arXiv:2509.09680*, 2025.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

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*, pp. 7514–7528, 2021.

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.

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

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

Nanshan Jia, Tingyu Zhu, Haoyu Liu, and Zeyu Zheng. Structured diffusion models with mixture of gaussians as prior distribution. *arXiv preprint arXiv:2410.19149*, 2024.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023.

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*, 2026.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

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

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

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.

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

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.

Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7086–7096, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Eliya Nachmani, Robin San Roman, and Lior Wolf. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

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.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

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

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 *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

L Rout, Y Chen, N Ruiz, C Caramanis, S Shakkottai, and W Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Hu0FSOSEyS.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023.

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.

Christoph Schuhmann and Romain Beaumont. Laion-aesthetics. *LAION. AI*, 2022.

Kihyuk Sohn, Lu Jiang, Jarred Barber, Kimin Lee, Nataniel Ruiz, Dilip Krishnan, Huiwen Chang, Yuanzhen Li, Irfan Essa, Michael Rubinstein, et al. Styledrop: Text-to-image synthesis of any style. *Advances in Neural Information Processing Systems*, 36:66860–66889, 2023.

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

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10738–10747. IEEE, 2022.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

Haofan Wang, Matteo Spinelli, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. Instantstyle: Free lunch towards style-preserving in text-to-image generation. *arXiv preprint arXiv:2404.02733*, 2024.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 2555–2563, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformer, 2024. URL https://arxiv.org/abs/2410.10629.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Xi Yu, Xiang Gu, Haozhi Liu, and Jian Sun. Constructing non-isotropic gaussian diffusion model using isotropic gaussian diffusion model for image editing. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2Ibp83esmb.

Mohsen Zand, Ali Etemad, and Michael Greenspan. Diffusion models with deterministic normalizing flow priors. *arXiv preprint arXiv:2309.01274*, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

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

Pascal Zwick, Nils Friederich, Maximilian Beichter, Lennart Hilbert, Ralf Mikut, and Oliver Bringmann. Lediflow: Learned distribution-guided flow matching to accelerate image generation. *arXiv preprint arXiv:2505.20723*, 2025.



Figure 11: **Text-to-Image Qualitative Examples**

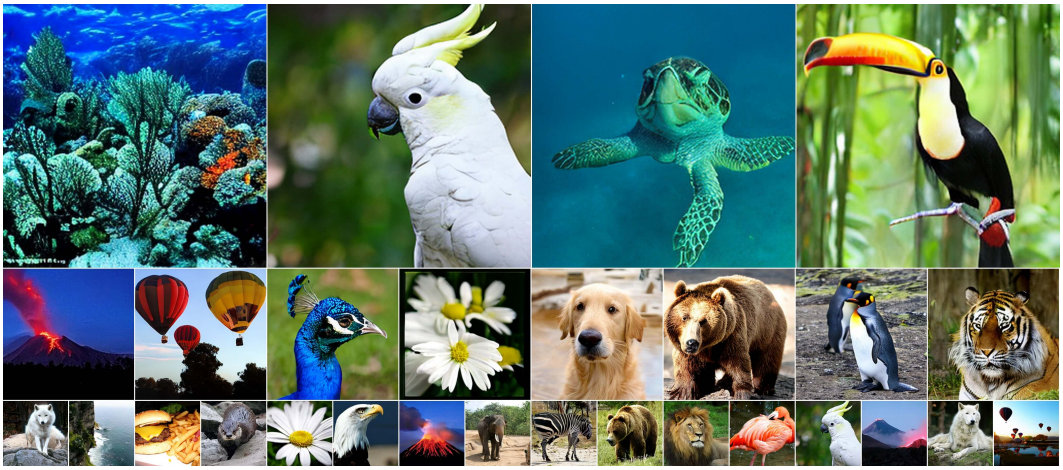


Figure 12: **Class-Conditional ImageNet Qualitative Examples**

A Related Works

A.1 Diffusion and Flow Models

Diffusion models are a class of generative models grounded in stochastic differential equations (SDEs). Their core principle is to gradually corrupt data through a stochastic forward process that adds noise over time, ultimately transforming the data distribution into a simple Gaussian distribution. This forward process is formally defined as

$$dx = f(x,t)dt + g(t)dW_t,$$

where $f(x,t)$ denotes the drift term, $g(t)$ is the diffusion coefficient, and dW_t represents the Wiener process, corresponding to infinitesimal Gaussian noise injected at time. The generative process corresponds to reversing this diffusion, aiming to recover the data distribution from noise. The reverse-time dynamics are given by

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dW_t,$$

where $\nabla_x \log p_t(x)$, known as the score function, captures the gradient of the log-density of the data distribution at time t. In practice, diffusion models learn this score function by training a neural network $s_\theta(x,t)$ via score matching:

$$\mathbb{E}_{t\sim U(0,T), x\sim p_t(x)}[\lambda(t)||\nabla_x \log p_t(x) - s_\theta(x,t)||^2],$$

where $\lambda(t)$ is a time-dependent weighting function.

Closely related to diffusion models are flow matching methods, which provide a framework for training Continuous Normalizing Flows (CNFs). Unlike diffusion models, which rely on stochastic SDE trajectories, flow matching constructs deterministic mappings between distributions via ordinary differential equations (ODEs). Specifically, the data transformation is governed by a learned vector field:

$$\frac{dx}{dt} = v_\theta(x, t),$$

where $v_\theta(x, t)$ parameterizes the velocity field. The model is trained by minimizing the discrepancy between the learned vector field and a target vector field $v_t(x)$:

$$\mathbb{E}_{t\sim U(0,T), x\sim p_t(x)}[|v_\theta(x, t) - v_t(x)|^2],$$

where $p_t(x)$ denotes the intermediate distributions along the transport path.

In contrast to diffusion models, flow matching leverages deterministic ODE trajectories rather than stochastic SDE dynamics, which can lead to improved computational efficiency and more stable training. As a result, flow matching can be viewed as a principled and efficient alternative to diffusion-based generative modeling.

B Experiments

B.1 Implementation details

All models use DiT-XL/2 (Peebles & Xie, 2023) as the backbone, operating on 32×32 latent representations obtained by encoding 256×256 images with a pretrained VAE (Stable Diffusion `sd-vae-ft-ema` (Rombach et al., 2022), downsample factor 8). Training is performed on $8\times$A100 GPUs using HuggingFace Accelerate (Gugger et al., 2022), with bfloat16 mixed precision and no gradient checkpointing. We optimize with AdamW (lr $= 10^{-4}$, weight decay $= 0$, $\varepsilon = 10^{-8}$) and a constant learning rate schedule. Gradient norms are clipped to 1.0 (ImageNet) and 2.0 (FFHQ). An exponential moving average (EMA) of model weights is maintained throughout training with decay $= 0.9999$; all evaluations use the EMA weights. Timesteps are sampled uniformly, and training follows the flow matching objective. Inference uses 100 Euler steps.

For StructFlow, segmentation masks are produced by SLIC superpixel oversegmentation (Achanta et al., 2012) precomputed at full image resolution (512×512) with 100 segments and compactness 30 for FFHQ, and 50 segments and compactness 50 for ImageNet. The structured noise is sampled at full resolution and downsampled to latent resolution (32×32) with standard deviation rescaling to preserve unit variance. The secondary noise strength λ (which controls intra-segment coherence) is annealed during training via a piecewise linear schedule rather than held fixed, gradually tightening the segment constraint as training progresses. For FFHQ, λ is annealed from 4.0 to 0.1 between steps 60k–100k and the model is trained for 130k steps total, using the 130k EMA checkpoint for evaluation. For ImageNet, StructFlow training is conducted in sequential stages initialized from a pretrained flow matching checkpoint, with λ annealed from 25.0 down to 4.0 and subsequently to 0.5 across approximately 3M total iterations, matching the budget of the baseline. The baseline DiT (FM-based) model is trained under an identical setup (same architecture, optimizer, batch size, training steps, and inference configuration) differing only in the noise sampling procedure, which uses standard isotropic Gaussian noise.

For the text-to-image experiments, we fine-tune SANA-1600M (Xie et al., 2024) (1024px) on the FLUX-Reason-6M dataset (Fang et al., 2025) using full fine-tuning (no LoRA). Training uses AdamW with lr $= 2\times10^{-5}$, $\beta = (0.9, 0.999)$, weight decay $= 10^{-4}$, and $\varepsilon = 10^{-8}$, with a constant learning rate schedule and no warmup. We use a global batch size of 32 (4×8 GPUs), bfloat16 mixed precision, no gradient checkpointing, and train for 100k steps. Caption dropout of 0.1 is applied to support classifier-free guidance at inference. Segmentation masks are produced on-the-fly via SLIC with 50 segments and compactness 40 at full resolution (1024×1024), then downsampled to latent resolution with std rescaling. As with ImageNet, the secondary noise strength λ is annealed from 22.0 to 0.5 between steps 5k–65k to progressively tighten the segment constraint during training.



Figure 13: **ImageNet Fine-Grained Local Resampling Qualitative Comparison.** Comparison across models trained with different λ_ς. Models trained with smaller λ_ς preserve unintended regions while producing diverse variations within the selected parts.

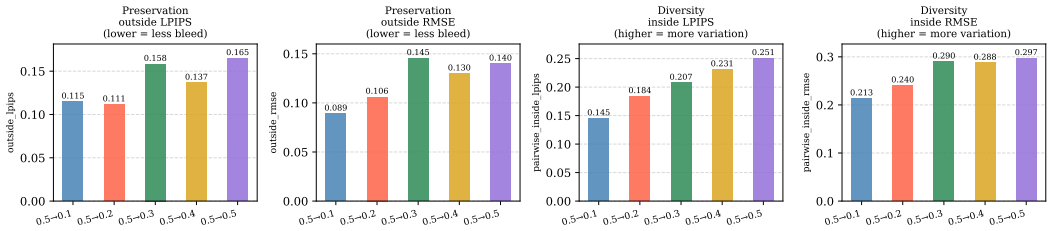


Figure 14: **ImageNet Fine-Grained Local Resampling Quantitative Comparison.** Quantitative results corroborate the qualitative findings: models trained with smaller λ_ς better preserve unaffected regions while maintaining diversity within the selected parts.

B.2 Qualitative and Quantitative Evaluation

In this section, we provide additional qualitative examples from a DiT trained with StructFlow and a SANA model post-trained using our method. Figures 11 and 12 illustrate results from these two models. As shown, both produce high-quality samples while exhibiting a range of emerging capabilities.

C Applications

C.1 Fine-Grained Image Editing and Part Resampling

In this section, we thoroughly evaluate the ability of fine-grained local resampling in StructFlow. This is the ability to locally re-synthesize a specific semantic region of an image while leaving the rest unchanged. Concretely, given a generated image and its associated oversegmentation mask, we identify a semantically meaningful subset of segments, regenerate only those segments under fresh noise, and measure both how well the unselected region is preserved and how diverse the regenerated variants are.

Procedure. For each evaluation sample, a base image is first synthesized together with a SLIC oversegmentation mask (Achanta et al., 2012) computed at the original image resolution. A region of interest is then identified automatically: we run CLIPSeg (Lüddecke & Ecker, 2022) on the base image using a text query, obtain a per-pixel relevance heatmap, average the heatmap values within each segment, and threshold the per-segment scores via Otsu's method (Otsu, 1979) (with a minimum score floor of 0.3) to obtain the set of selected segment IDs. For ImageNet models the query is the ground-truth class name of the conditioning label (e.g., *"golden retriever"*), so the selected region reflects the primary object of interest. For FFHQ faces the query is fixed to *"hair"*. Given the selected region mask, we produce $K = 5$ resample variants: each variant uses the same primary noise as the base image for the unselected pixels, but draws fresh secondary noise only inside the selected region, so that all stochasticity outside it is held fixed. This directly mirrors the structured noise decomposition of our StructFlow training and provides a clean isolation of the regenerated zone.

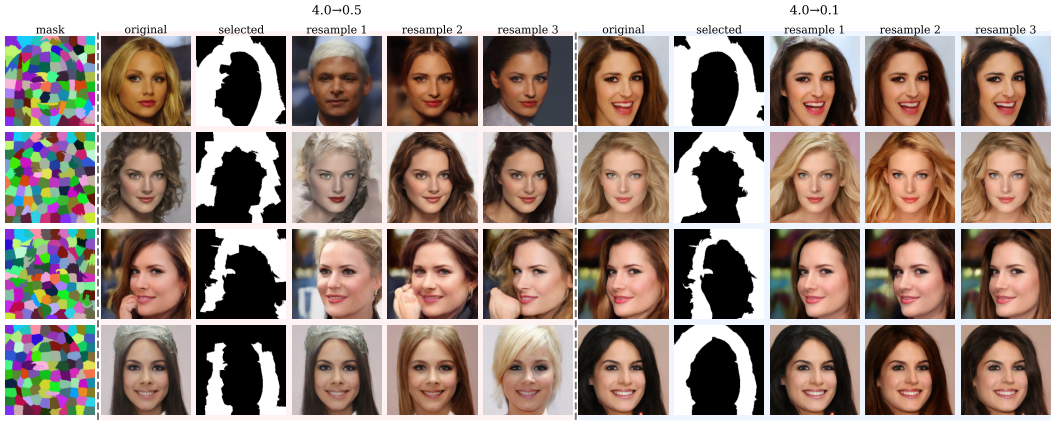


Figure 15: **FFHQ Fine-Grained Local Resampling Qualitative Comparison.** Comparison across models trained with different λ_ς. Models trained with smaller λ_ς preserve unintended regions while producing diverse variations within the selected parts.

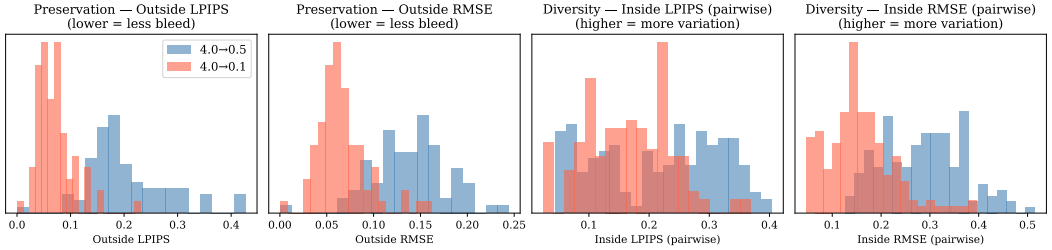


Figure 16: **FFHQ Fine-Grained Local Resampling Quantitative Comparison.** Quantitative results corroborate the qualitative findings: models trained with smaller λ_ς better preserve unaffected regions while maintaining diversity within the selected parts.

Metrics. We measure two complementary properties across the K resamples: We propose two complementary metrics to evaluate this behavior: **Preservation** (lower is better) measures how much the *unselected* region changes between the base image and each resample. We report the mean pixel-level RMSE (*Outside RMSE*) and perceptual LPIPS distance (*Outside LPIPS*) (Zhang et al., 2018), computed over unselected pixels only. **Diversity** (higher is better) measures how much the *selected* region varies *across* resamples. We compute all $\binom{K}{2}$ pairwise comparisons within the selected region and report the mean pairwise RMSE (*Inside RMSE*) and LPIPS (*Inside LPIPS*). A well-behaved model should achieve both low preservation error (minimal disturbance to surrounding regions) and high diversity (meaningful variation within the edited region).

ImageNet Results. We evaluate five model variants, each fine-tuned from a shared checkpoint that was originally trained with $\lambda_\varsigma = 0.5$, continuing with a cosine schedule that decays λ_ς to a final value of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ respectively (no mixed-coherence is used in these experiments). Evaluation is performed on 16 class-conditional validation samples with classifier-free guidance scale 4.0; the CLIPSeg query matches the class label of each sample. As shown in Figures 13 and 14, models trained to smaller final λ_ς exhibit substantially better preservation of the unselected region, with only a modest reduction in inside diversity. This confirms the theoretical expectation: tighter coupling between pixels of the same segment (smaller secondary noise) produces cleaner segment boundaries and more faithful outside preservation during resampling, while still leaving enough degrees of freedom

FFHQ. Results We compare two unconditional face generation models that differ in their secondary noise schedule: one trained with λ_ς annealed from 4.0 to 0.5, and one with a more aggressive schedule from 4.0 to 0.1. Evaluation uses 16 real images with SLIC masks, and the query *"hair"* is used for automatic region selection. Results in Figures 15 and 16 show the same trend as ImageNet: the model

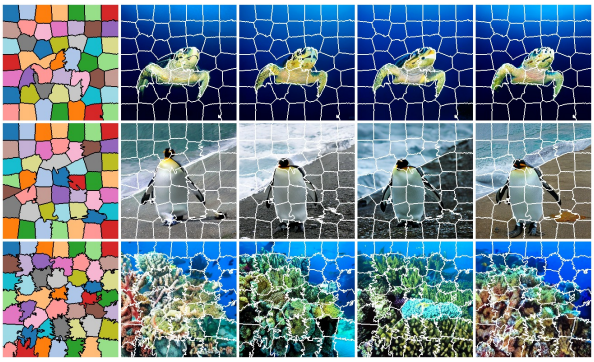


Figure 17: **Structure Preserving Generation.** StructFlow allows generation of diverse samples using the same grounding superpixel masks while preserving the overall structure.

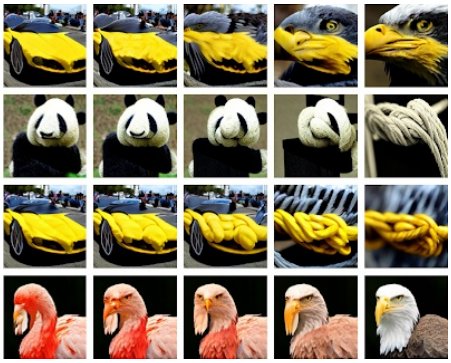


Figure 18: StructFlow's structure preserving capabilities allows interesting class interpolation during the early denoising steps of the generation.

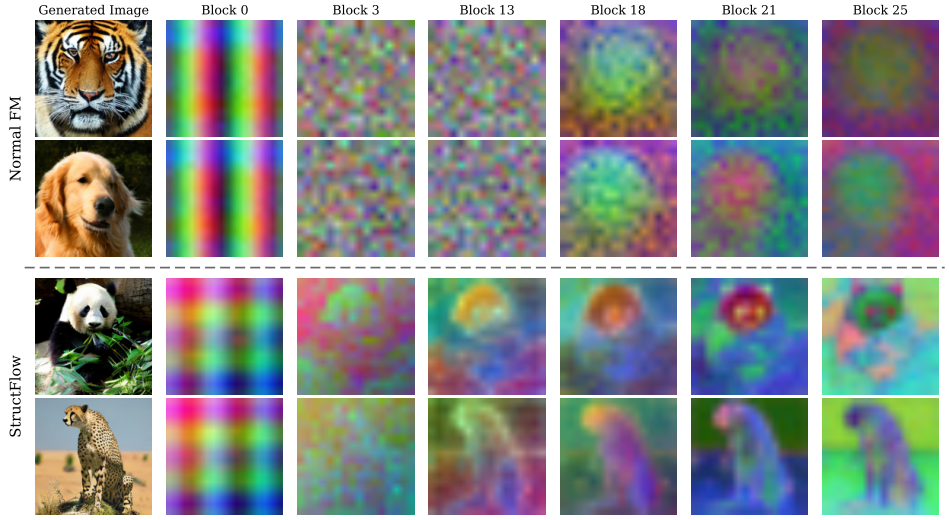


Figure 19: **Informative Early Features.** StructFlow exhibits spatially coherent features from the earliest denoising steps, while Normal FM remains largely unstructured.

with the smaller final λ_ς (4.0→0.1) substantially outperforms on preservation, indicating that the hair region can be re-synthesized with minimal leakage into the face and background, while the diversity of the generated hair variants remains adequate for practical use.

C.2 Structure Preserving Generation

In this section, we provide additional qualitative examples showing that StructFlow naturally adheres to the provided segment boundaries, producing generations that preserve the underlying spatial structure. This structural consistency also enables class interpolation, where changing the class condition during early denoising alters the semantic content while largely maintaining the same layout. Figures 17 and 18 show additional results.

C.3 Early Emergence of Informative Features

We compare the intermediate representations of two DiT-XL/2 models trained on ImageNet 256×256 and sampled with 100 Euler flow-matching steps using classifier-free guidance (scale $= 4.0$): (i) a standard flow-matching baseline, referred to as Normal FM, and (ii) our StructFlow model, whose source noise is constructed using SLIC oversegmentation masks with 50 segments. For each model,

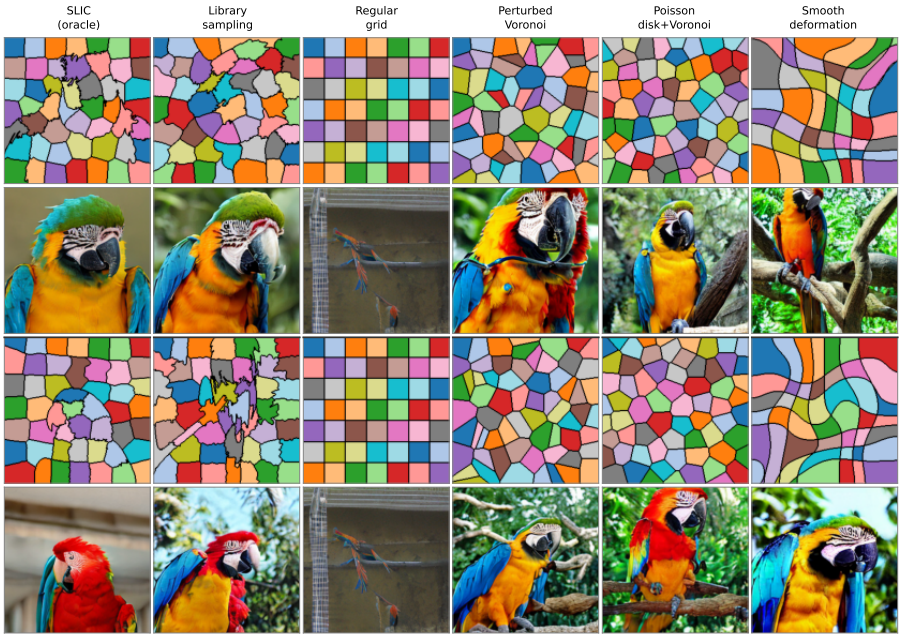


Figure 20: Comparison of training-free mask generation strategies

we generate $N = 60$ images from 12 visually related animal classes, with 5 samples per class. These classes are chosen because they share common semantic parts, such as eyes, ears, muzzles, and fur patterns, making cross-sample feature alignment easier to inspect in the PCA maps. We extract features from all transformer blocks at five denoising fractions: 10%, 30%, 50%, 70%, and 90% of the sampling trajectory. For each *(block, timestep)* pair, we pool the spatial tokens from all 60 generated images and fit a 3-component PCA. We then project each token onto the three principal components, reshape the projected features back to the spatial grid, and upsample them to the image resolution. The three PCA components are visualized as RGB channels and normalized using the global minimum and maximum across all samples for that *(block, timestep)* pair. Therefore, within each pair, the same color corresponds to the same feature direction across all samples, making cross-image consistency directly visible. Since PCA is fit independently for each *(block, timestep)* pair, colors should not be compared across different rows or columns of the figure.

Figure 19 shows PCA maps from an early denoising stage, at 10% of the trajectory, i.e., after only 10 of the 100 sampling steps. We visualize six representative transformer blocks. The first two rows correspond to Normal FM generations, while the bottom two rows correspond to StructFlow generations. The first column shows the final generated image for reference. Although the latent state is still highly noisy at this stage, the PCA maps of StructFlow already show clear spatial organization across multiple blocks, including coarse object boundaries, foreground–background separation, and rough object layout. In contrast, the Normal FM maps at the same timestep remain largely unstructured, with colors varying irregularly across patches rather than forming coherent regions. For Normal FM, comparable spatial organization appears only later in the denoising process, around 50%–90% of the trajectory. This suggests that the structured source used by StructFlow encourages the model to organize its internal features around spatially meaningful regions from the earliest sampling steps.

D TEXT-TO-IMAGE STRUCTFLOW

Details of the StructFlow text-to-image training setup are provided in Sec. B.1. Additional qualitative examples are shown in Fig. 11.

E MASK GENERATION STRATEGIES

As discussed in Section 4.5, we explore several strategies for generating segmentation masks at inference time. Here, we provide a more detailed discussion of these approaches.

Lightweight generative model. A straightforward option is to train a small generative model to produce plausible superpixel layouts on demand. We implement this as a compact U-Net flow-matching model (14.1M parameters) conditioned on the ImageNet class label, trained on 1.28M precomputed SLIC masks in approximately 3 hours on two A100 GPUs. The model closely matches real SLIC statistics in segment count, mean segment size, and spatial autocorrelation. Crucially, mask generation costs only 7–35 ms per sample (batch sizes 32–128, 20 inference steps), roughly two orders of magnitude faster than the main DiT image generation pipeline, making it a negligible overhead in practice.

Training-free approaches. We explore five training-free alternatives requiring no additional model. The simplest is *library sampling*: at inference time, a mask is drawn uniformly at random from the pool of precomputed SLIC masks. Since SLIC superpixels with high compactness are governed primarily by spatial regularity rather than image content, a mask from an unrelated image still provides a plausible spatial layout. This approach is zero-cost and produces in-distribution masks by construction.

The remaining strategies construct masks procedurally. A *regular grid* partitions the image into uniform rectangular cells. *Perturbed Voronoi* tessellation places seeds on a regular grid and applies isotropic Gaussian noise before computing the nearest-neighbour Voronoi diagram; the perturbation magnitude controls the degree of organic irregularity. *Poisson-disk Voronoi* removes the grid prior entirely: seeds are placed using Bridson's algorithm (Bridson, 2007), which enforces a minimum inter-seed distance and produces blue-noise point distributions with no underlying grid memory, yielding more organic, irregular cells while still guaranteeing uniform spatial coverage. Finally, *smooth deformation* starts from a regular $n \times n$ grid and applies a spatially smooth random displacement field (obtained by blurring independent Gaussian noise with a large kernel) to warp the grid boundaries.

Figure 20 compares example masks and generated images for all six strategies. As the figure illustrates, the regular grid and smooth deformation approaches produce masks that are noticeably out of distribution relative to the SLIC superpixels the model was trained on: their large, rectangular or broadly warped regions differ substantially in shape, size distribution, and boundary character from organic SLIC segments, and this mismatch leads to visibly degraded generation quality. By contrast, the Voronoi-based approaches (both perturbed Voronoi and Poisson-disk Voronoi) produce masks whose local statistics are much closer to SLIC, and the resulting generations are correspondingly higher quality and more coherent. Library sampling inherits the exact SLIC distribution by construction and therefore performs comparably to the oracle. These results confirm that matching the mask distribution seen during training is the key factor for high-quality generation, and that Voronoi-based procedural masks provide a practical and effective training-free alternative to precomputed SLIC masks.

F ABLATIONS

We conduct a small ablation to study the effect of the number of local regions used for structured noise construction, controlled by the number of SLIC segments. Specifically, we train three DiT models with 25, 50, and 75 segments, respectively, while keeping all other training settings fixed. Each model is trained for 1.8M iterations. Figure 21 shows qualitative samples from the final checkpoints. Overall, we observe no major qualitative difference across these settings: all three models produce reasonable samples and preserve the main benefits of structured noise. However, the number of segments still provides a useful practical trade-off. Using too many segments can make the source structure overly fine-grained, which may restrict the model and reduce its ability to generate details that extend across segment boundaries. Conversely, using too few segments weakens the spatial control provided by the structured source. Based on this trade-off, we use 50 segments as a balanced default in our experiments.



Figure 21: **Effect of segment count.** Varying the number of SLIC segments produces similar qualitative results, with 50 segments providing a balanced default.

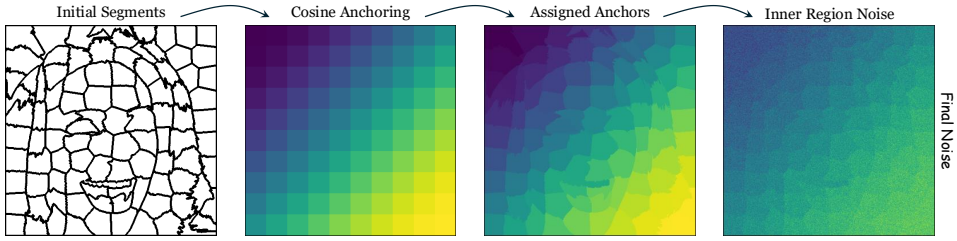


Figure 22: **Cosine-Induced Spatial Anchoring and Noise Construction.** Segment anchors are deterministically sampled from a fixed cosine grid, after which inner-region noise is added to construct the final structured source sample.

We also explored replacing superpixels with semantic segmentation masks. In particular, for FFHQ, we used face-part segmentations, such as eyes, hair, lips, and other facial regions, as the local regions for structured noise construction. This setting was trainable and produced reasonable samples. However, we found it less suitable for our goal for two main reasons. First, semantic regions do not always correspond to local coherence: a large region such as hair can cover spatially distant or visually diverse areas, where the pixels are not necessarily strongly correlated. Second, semantic masks provide much coarser control than superpixels, since each region often corresponds to an entire object part rather than a fine-grained local area. construction.

G Comparison with Structure Preserving and Local Editing Baselines

We compare StructFlow with methods specifically designed for structure-preserving generation and local editing. Our goal is not to claim state-of-the-art performance on these tasks; rather, these capabilities emerge out of the box from the proposed structured source distribution, without any task-specific training. Nevertheless, we compare against dedicated baselines and show that StructFlow achieves comparable performance. All StructFlow results use the SANA model post-trained with our method (Section 4.4).

G.1 Baselines

For *structure preserving*, we compare against ControlNet-HED (Zhang et al., 2023; Xie et al., 2024), ControlNet-Canny (Zhang et al., 2023; Podell et al., 2024), T2I-Adapter-Canny (Mou et al., 2024), and SDEdit (Meng et al., 2021). ControlNet-HED and SDEdit use SANA, while the Canny-based baselines use SDXL. We additionally report Base SANA, sampled from standard Gaussian noise without structural conditioning. For *local editing*, we compare against training-free methods, SDEdit (Meng et al., 2021) and Blended Latent Diffusion (Avrahami et al., 2023); dedicated inpainting models, SD-1.5 Inpainting (Rombach et al., 2022) and SDXL Inpainting (Podell et al., 2024); and

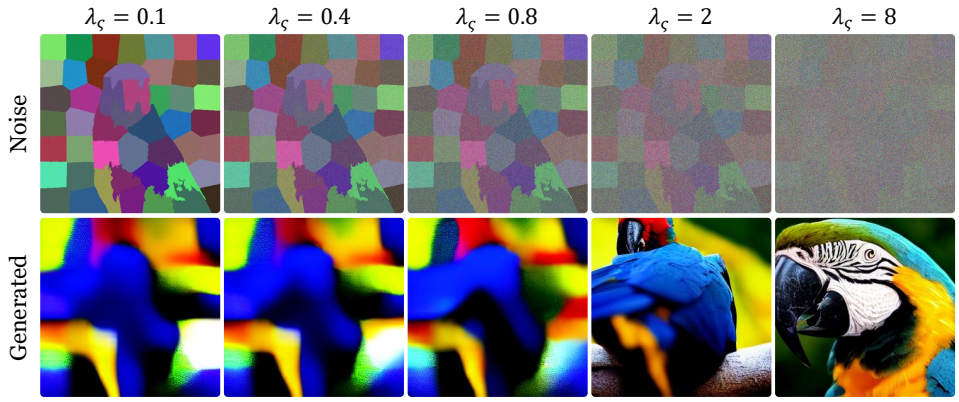


Figure 23: **Effect of λ_ς.** Noise constructed with varying λ_ς and evaluated using a standard FM-based DiT. As $\lambda_\varsigma \to \infty$, the noise approaches i.i.d. Gaussian, matching the training distribution of the base model and yielding correct generation. This enables post-training via Progressive Coherence Annealing, by initializing with large λ_ς and gradually reducing it to introduce structure.

instruction-based editors, InstructPix2Pix (Brooks et al., 2023) and FLUX.1-Kontext (Labs et al., 2025). We report both StructFlow part resampling and StructFlow + Blending, where the latter additionally anchors the region outside the edit mask during sampling.

G.2 Evaluation Metrics

For *structure preserving*, we measure structure adherence using DINO self-similarity distance (Tumanyan et al., 2022; 2023) and depth RMSE using Depth-Anything-V2 (Yang et al., 2024). We additionally report LPIPS (Zhang et al., 2018) for diversity, CLIPScore (Hessel et al., 2021) for text alignment, and CLIP-IQA (Wang et al., 2023) for image quality. For *local editing*, following PIE-Bench (Ju et al., 2023), we measure DINO structure distance and background preservation using PSNR and LPIPS outside the edited region. Diversity is measured using pairwise LPIPS within the edited region, and image quality using CLIP-IQA.

G.3 Evaluation Protocol

For *structure preserving*, we evaluate on 64 PartiPrompts (Yu et al., 2022) prompts spanning categories with meaningful spatial layouts. For each prompt, Base SANA generates a reference image, from which we extract the corresponding condition for each method: SLIC segments for StructFlow, HED or Canny edges for the respective conditional models, and the reference image itself for SDEdit. Each method generates five samples per prompt, resulting in 320 images per method. For *local editing*, we construct the benchmark from PIE-Bench (Ju et al., 2023), retaining 70 examples with localized target objects. StructFlow source images are generated using SLIC masks, and edit regions are identified with CLIPSeg (Lüddecke & Ecker, 2022). All methods edit the same source image and use the same mask whenever applicable. For instruction-based editors, we use the fixed instruction *"Replace the ⟨object⟩ with a different ⟨object⟩, keeping everything else unchanged."* Each method generates four variants per example, resulting in 280 images per method.

G.4 Findings

Structure preserving. Table 2 shows that StructFlow substantially improves structure adherence over Base SANA while maintaining high diversity, text alignment, and image quality. Methods explicitly conditioned on edges or the reference image can achieve stronger structure adherence, but generally at the cost of diversity or prompt flexibility. Overall, StructFlow provides a competitive trade-off despite not being trained specifically for structure preservation. Qualitative comparisons in Figures 8 and 24 further show that StructFlow preserves the overall layout while remaining flexible to semantic changes in the prompt.

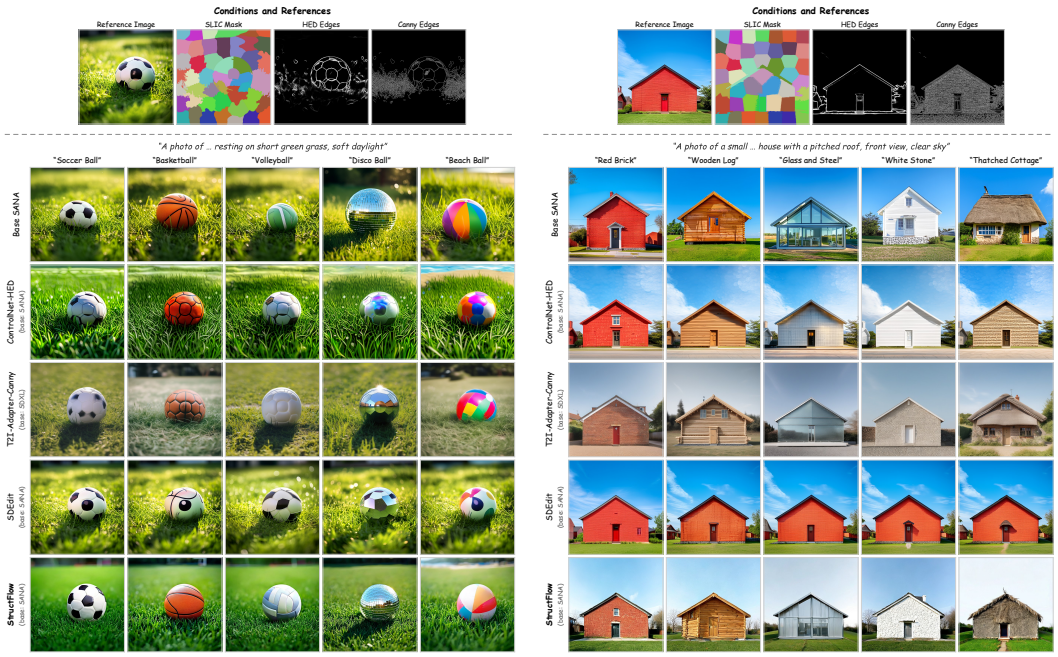


Figure 24: Qualitative comparison of structure-preserving methods.

Local editing. Table 3 shows that StructFlow preserves background and structure better than SDEdit while maintaining comparable region diversity. Dedicated inpainting and blending methods achieve stronger background preservation, as expected from their explicit spatial constraints. When combined with blending, StructFlow achieves the strongest structure and background preservation among the evaluated methods while retaining competitive image quality and diversity.

H LIMITATIONS

While StructFlow introduces a new paradigm for source distribution design that improves localized controllability and structure preservation, it also has several limitations. First, our method relies on segmentation masks or mask proxies to construct the structured source distribution. Although we explore retrieval-based and training-free alternatives at inference time, generation quality can depend on how closely these masks match the distribution used during training. Second, the structured source introduces additional hyperparameters, such as the number of segments and the secondary noise scale λ_ς, which control the strength of locality and may require tuning across datasets, resolutions, or model architectures.