Flow-Optimizer: Revealing an Optimizable Flow Latent Space via One-Step Inversion for Controlled Interpolation and Editing

Yan Zheng
University of Texas at Austin
yanzheng@utexas.edu

Yi Yang
University of Edinburgh
arloyang397@gmail.com

Abstract

*We introduce **Flow-Optimizer**, a novel framework that leverages the unique oscillatory behavior observed in rectified flow-based models to enable high-quality image interpolation and editing. While existing methods often struggle to preserve semantics and structure, our approach reveals that latent oscillations form semantically coherent clusters. These clusters can be further optimized for smooth and controllable transformations, making it ideal for tasks such as structure-preserving image interpolation. This insight provides a new perspective on the latent space of flow models and enables practical techniques for structure-aware, interpretable image editing. Extensive Experiments show clear improvements in visual quality, consistency, and efficiency over prior methods.*

1. Introduction

Flow-based models provide a complementary generative modeling approach that addresses some of these limitations. Unlike diffusion models, flow-based models parameterize an invertible, deterministic transformation between a simple latent distribution (often Gaussian) and complex data distributions, enabling efficient and exact likelihood evaluation and sampling. By harnessing invertible neural networks, flow models facilitate direct manipulation of latent variables, thus offering more explicit control over generated content. Recent research integrating flow-based frameworks into diffusion-inspired pipelines has demonstrated improved efficiency and controllability, highlighting promising directions for enhanced generative modeling in images, videos, and 3D structures.

In this paper, we investigate the problem of using large pre-trained flow model for image interpolation, where source and target images share a similar layout (e.g., depth- or mask-aligned), is a challenging yet essential task in computer vision. Unlike traditional morphing methods that handle structural differences, we explore an orthogonal direction by leveraging the refined reverse stages of flow-based generative models to achieve smooth interpolation, focusing on fine-grained details such as color, style, and texture under a consistent structural layout.

Our approach reformulates interpolation as an optimization problem in the latent space of a rectified flow model[1, 4, 5]. Building on recent advances in oscillation inversion techniques[9], we propose a one-step inversion strategy that reveals a local, optimizable latent space. Rather than yielding a single deterministic solution[7], we found that the fixed-point iteration in oscillation inversion produces symmetric latent clusters. By averaging the latent codes within these clusters, we obtain a stable initialization that effectively captures the integrated posterior mass in a local neighborhood, providing the optimal starting point for subsequent optimization.

After identifying this stable latent initialization, we further explore how it can guide downstream transformations. We observe that the latent space around this initialization exhibits local smoothness—allowing us to perturb the reconstruction target without disrupting optimization stability. This insight enables us to extend beyond point-wise reconstruction toward continuous interpolation. Specifically, we formulate interpolation as a sequence of latent-space optimization steps, where the target is gradually perturbed. At each stage, we reinitialize the process using the averaged latent from the previous step, ensuring that the path remains within a stable region. This iterative strategy prevents latent drift and enables smooth transitions across fine-grained attributes while preserving structure and identity. Our contributions are summarized as follows:

- We propose a novel optimization framework that utilizes a one-step inversion derived from oscillation inversion methods to obtain a local optimizable latent space.
- We investigate leveraging refined reverse stages of flow-based models for smooth interpolation in settings where images share similar structural layouts.
- We design an iterative mechanism that ensures a smooth and stable optimization trajectory from source to target, effectively mitigating latent deterioration.

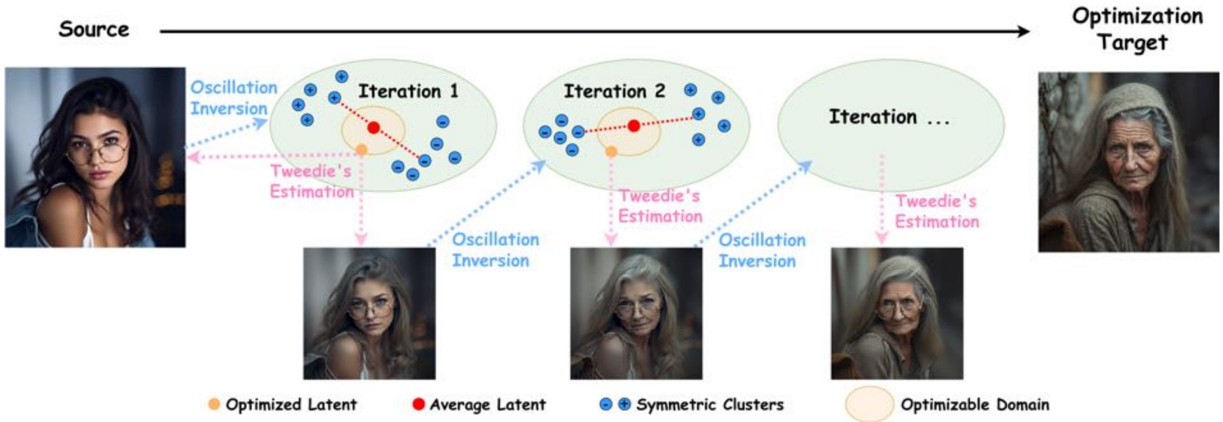

Figure 1. *Flow-Optimizer* leverages oscillation inversion to generate symmetric clusters (blue dots) from fixed-point iterations. These clusters naturally separate into odd and even iterations, allowing us to compute average latent points (red dots). We subsequently obtain optimized points (orange) based on this initialization. Using Tweedie's estimation to approximate ODE paths, we iteratively transform the source image through a sequence of smooth transitions, gradually approaching the target with high-quality intermediary results.

2. Method

2.1. Preliminary: Rectified Flow

Rectified Flow [1, 4, 5] is a generative approach that enables smooth distribution transitions via ODEs. Given $\mathbf{Z}_0 \sim \pi_0$ and $\mathbf{Z}_1 \sim \pi_1$, the transition follows a linear interpolation $\mathbf{Z}_t = (1-t)\mathbf{Z}_0 + t\mathbf{Z}_1$ for $t \in [0,1]$. To preserve the marginal distribution of $\mathbf{Z}_t$, the following ODE is used:

$$\frac{d\mathbf{Z}_t}{dt} = v(\mathbf{Z}_t, t), \tag{1}$$

where v represents the velocity field. In practice, v is parameterized by a neural network $v_\theta(\mathbf{Z}_t, t)$ and optimized via stochastic coupling $(\mathbf{Z}_0, \mathbf{Z}_1) \sim (\pi_0, \pi_1)$ and $t \sim \text{Uniform}([0,1])$:

$$v_\theta(\mathbf{Z}_t, t) = \arg\min_\theta \mathbb{E}\left[\left\|(\mathbf{Z}_1 - \mathbf{Z}_0) - v(\mathbf{Z}_t, t)\right\|^2\right] \tag{2}$$

where $\mathbf{Z}_t = (1-t)\mathbf{Z}_0 + t\mathbf{Z}_1$.

2.2. Problem Definition

Following [5], the ODE path z_t preserves the marginal distribution from $\mathbf{Z}_0 \sim \pi_0$, typically a standard Gaussian. While many ODE inversion methods seek an exact solution, our goal is to identify a locally optimizable latent space rather than a precise inversion. We propose that integrated posterior mass offers a more robust initialization than pointwise posterior mass:

$$z^*_{t_i} = \arg\max_{z_{t_i}} \int_{\mathcal{N}(z_{t_i})} p(z' \mid y)\, dz' \tag{3}$$

$$= \arg\max_{z_{t_i}} \int_{\mathcal{N}(z_{t_i})} C \cdot p(y \mid z_{t_i})\, p(z_{t_i}) \tag{4}$$

$$\text{where} \quad \mathcal{N}(z_{t_i}) = \{z' : \|z' - z_{t_i}\| \leq \epsilon\}. \tag{5}$$

Even in deterministic models, latent inversion can be framed probabilistically: given a real-world image y, we seek $\mathbf{Z}_{t_i}$ that maximizes the integrated mass.

2.3. Optimized Inversion Starting Point

Directly solving Eq. (4) is computationally expensive due to the need for backpropagation through the ODE. To address this, we adopt Tweedie's formula [2] to approximate the ODE path:

$$\arg\min_{z_{t_i}} \sum_{z' \in \mathcal{N}(z_{t_i})} \|y - z' - (\sigma_0 - \sigma_i)v_\theta(z_{t_i}, t_i)\|^2, \tag{6}$$

where σ_i is the scheduled coefficient at timestep i.

Though exact inversion approach can get inverted latent with good reverse path. They still incurs a non-negligible error with this Tweedie's estimation. Moreover, our formulation aims to find the point where the entire neighborhood closely matches the target y, merely inversion method with numerical solver or path mixture often results in an unstable solution that lacks smooth differentiability. To this end, we propose a novel method to approximate Eq. (6). Following [9], the fixed-point sequence for the one-step inversion problem in large flow models exhibits symmetric clusters, with pairwise means concentrating in a stable region. Within this region, the one-step reward path provides a reliable approximation of the original input. We define the fixed-point iteration as:

$$z^{(k+1)}_{t_0} = y - (\sigma_0 - \sigma_i)v_\theta(z^{(k)}_{t_0}, \sigma_{t_0}), \tag{7}$$

with the initial condition $z^{(0)}_{t_0} = y$.

Due to the symmetric clustering, latent variables at odd and even iterations separate naturally:

$$\mathcal{Z}_{\text{odd}} = \{z^{2k+1}_{t_0}\} \quad \text{and} \quad \mathcal{Z}_{\text{even}} = \{z^{2k}_{t_0}\}.$$

We further observe that averaging corresponding odd and

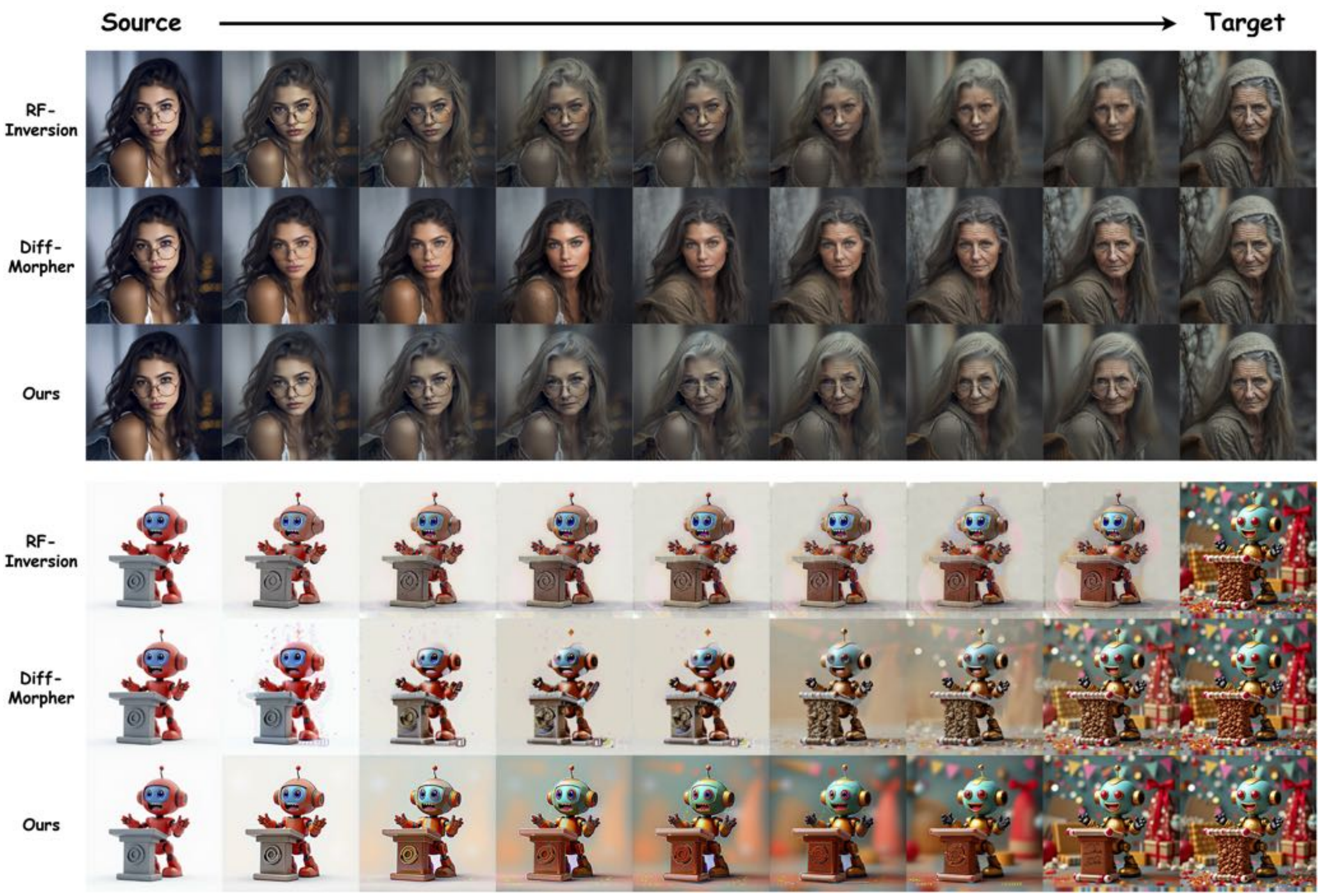

Figure 2. *Top:* Age progression transformation from young to elderly. *Bottom:* Scene transformation of a toy robot with increasingly elaborate background. Compared to RF-Inversion[6] and Diff-Morpher[8], our method *Flow-Optimizer* produces significantly smoother transitions and better preserves identity and structural consistency.

even variables,

$$\bar{z} = \frac{z_{\text{odd}} + z_{\text{even}}}{2},$$

yields a sufficiently accurate approximation of y under Tweedie's estimator. Thus, by randomly pairing elements from $\mathcal{Z}_{\text{odd}}$ and $\mathcal{Z}_{\text{even}}$, we construct latent variables at t_0 that effectively lower the objective in Eq. (6). Next, we randomly select one averaged latent—akin to Monte Carlo sampling in a favorable neighborhood—to initialize optimization.

2.4. From Reconstruction to Smooth Interpolation

While solving Eq. (6) for reconstruction, we observe that the smooth structure around the averaged latent $\bar{z}$ reliably provides an initial point for reconstructing y. This not only stabilizes optimization but also suggests that the surrounding latent space follows a well-behaved trajectory, allowing smooth and controlled adjustments.

Building on this, we find that optimization extends beyond reconstruction to broader tasks, such as controlled interpolation. Specifically, replacing y with a perturbed version $\tilde{y}$ still yields high-quality latents within a few steps. This reveals a natural interpolation path between y and $\tilde{y}$, enabling smooth transitions between input variations. Formally, we optimize:

$$\arg\min_{z_{t_i}} \sum_{z' \in \mathcal{N}(z_{t_i})} \|\tilde{y} - z' - (\sigma_0 - \sigma_i) v_\theta(z_{t_i}, t_i)\|^2. \quad (8)$$

To ensure the entire interpolation path remains within a valid region without latent deterioration, we propose an iterative optimization pipeline. We begin by denoting the initial target as y_0 and optimizing towards y_0 to obtain a sequence of latent estimates via Tweedie's estimator:

$$z(0,0),\ z(0,1),\ \ldots,\ z(0,k).$$

The final latent variable $z(0,k)$ is then treated as the new target y_1 to initiate the next optimization segment. This process is iteratively repeated, ensuring that each segment of the interpolation path remains within a well-behaved region of the latent space.

Figure 3. Guided Editing with mixture loss towards 'old' and 'robot'

Figure 4. Guided Editing from Game Rendering style towards photorealistic style

$$\text{Loss}_1 = \sum_{z' \in \mathcal{N}(z_{t_i})} \left\| \tilde{y}_1 - z' - (\sigma_0 - \sigma_i)\, v_\theta(z_{t_i}, t_i) \right\|^2. \quad (9)$$

We observe that these individual interpolation targets can be combined into a composite loss function:

$$\text{Loss} = \text{Loss}_1 + \text{Loss}_2 + \cdots , \quad (10)$$

leading to a more flexible approach that can accommodate a wide range of downstream tasks, as guided elaborate editing towards mixture of semantics in Figure 3 and sim-to-real editing in Figure 4.

3. Experiments

We evaluate *Flow-Optimizer* across diverse image manipulation tasks. Results show that it enables high-quality, controllable interpolation while preserving semantic consistency and structure, and significantly improves visual quality and efficiency over existing methods.

3.1. Implementation Details

All experiments were conducted using the Black Forest Labs' FLUX.1-Depth-dev model[3], a strong foundation for depth-aware image generation. We used a single A6000 GPU with 48G memory for all experiments. For baseline comparison, we implemented all the baselines we are considering.

For oscillation inversion, we used 30 fixed-point iterations, which empirically sufficient to capture stable clusters. A timestep of $t_i = 0.55$ was chosen to balance structure preservation and edit-ability. Optimization was performed using Adam with learning rates of 0.05 for the noise predictor and 0.01 for the latent variable. Our iterative strategy used 5–6 rounds with 4–5 steps each. The total time for a complete interpolation sequence is about 10 seconds, significantly faster than comparable methods requiring minutes.

3.2. Geometry-Preserving Image Interpolation

Geometry-preserving interpolation is challenging for existing methods, as it requires preserving structure while allowing appearance to vary smoothly. Although image generation conditioned on depth, masks, and sketches can yield visually plausible, well-aligned images, it still lacks fine-grained control, which hinders the direct use of generative editing techniques in downstream tasks such as texture synthesis, relighting, and inverse rendering. *Flow-Optimizer* addresses this through two key components: (1) oscillation-based inversion, which discovers semantically coherent latent clusters that maintain structure, and (2) an optimization formulation that explicitly promotes structural consistency during interpolation.

Figure 2 demonstrate *Flow-Optimizer*'s effectiveness on elaborate character transformations. Our method captures simultaneous changes in textures and lighting while preserving the underlying object features and geometry. This contrasts sharply with RF-Inversion[6], which struggles to create meaningful intermediate states, and Diff-Morpher[8], which produces inconsistent transitions with varying rates of change across the sequence.

To support our claim that our method achieves more consistent interpolation regarding especially underlying geometry, we first collect 20 face pairs generated by rough depth-aligned generation, and use our method, RF-Inversion, as well as diff-morpher to interpolate. We then use Google Media Pipe toolkit to extract 478 key features per image to track the interpolation consistency in geometry. The average feature point distance of our interpolation result is 85.33, compared to 123.74 for RF-inversion and 153.22 for Diff-morpher. Quantitative results support our findings. *Flow-Optimizer* outperforms prior methods.

4. Conclusions

In this paper, we presented *Flow-Optimizer*, a framework that leverages the intrinsic structure of rectified flow models' latent space for high-quality image interpolation. By combining oscillation inversion techniques with optimized initialization point selection and an iterative optimization strategy, we effectively address the challenges of maintaining semantic consistency during transformations. Our experimental results demonstrate that *Flow-Optimizer* outperforms existing methods in visual quality and structural preservation. This work not only provides practical techniques for structure-aware image editing but also offers new insights into the fundamental properties of flow model latent spaces.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *ICLR*, 2023. 1, 2

[2] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 2011. 2

[3] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 4

[4] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023. 1, 2

[5] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ICLR*, 2023. 1, 2

[6] L Rout, Y Chen, N Ruiz, C Caramanis, S Shakkottai, and W Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *ICLR*, 2025. 3, 4

[7] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *ICLR*, 2025. 1

[8] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. Diffmorpher: Unleashing the capability of diffusion models for image morphing. In *CVPR*, 2024. 3, 4

[9] Yan Zheng, Zhenxiao Liang, Xiaoyan Cong, Yuehao Wang, Peihao Wang, Zhangyang Wang, et al. Oscillation inversion: Understand the structure of large flow model through the lens of inversion method. *arXiv preprint arXiv:2411.11135*, 2024. 1, 2