This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

BiFM: Bidirectional Flow Matching for Few-Step Image Editing and Generation

Yasong Dai[1,2], Zeeshan Hayder[1,2], David Ahmedt-Aristizabal[2], Hongdong Li[1,3]
[1]Australian National University, [2]CSIRO Data61, [3]Amazon
{yasong.dai, hongdong.li}@anu.edu.au
{zeeshan.hayder, david.ahmedtaristizabal}@data61.csiro.au

Abstract

Recent diffusion and flow matching models have demonstrated strong capabilities in image generation and editing by progressively removing noise through iterative sampling. While this enables flexible inversion for semantic-preserving edits, few-step sampling regimes suffer from poor forward process approximation, leading to degraded editing quality. Existing few-step inversion methods often rely on pretrained generators and auxiliary modules, limiting scalability and generalization across different architectures. To address these limitations, we propose BiFM (Bidirectional Flow Matching), a unified framework that jointly learns generation and inversion within a single model. BiFM directly estimates average velocity fields in both "image → noise" and "noise → image" directions, constrained by a shared instantaneous velocity field derived from either predefined schedules or pretrained multi-step diffusion models. Additionally, BiFM introduces a novel training strategy using continuous time-interval supervision, stabilized by a bidirectional consistency objective and a lightweight time-interval embedding. This bidirectional formulation also enables one-step inversion and can integrate seamlessly into popular diffusion and flow matching backbones. Across diverse image editing and generation tasks, BiFM consistently outperforms existing few-step approaches, achieving superior performance and editability.

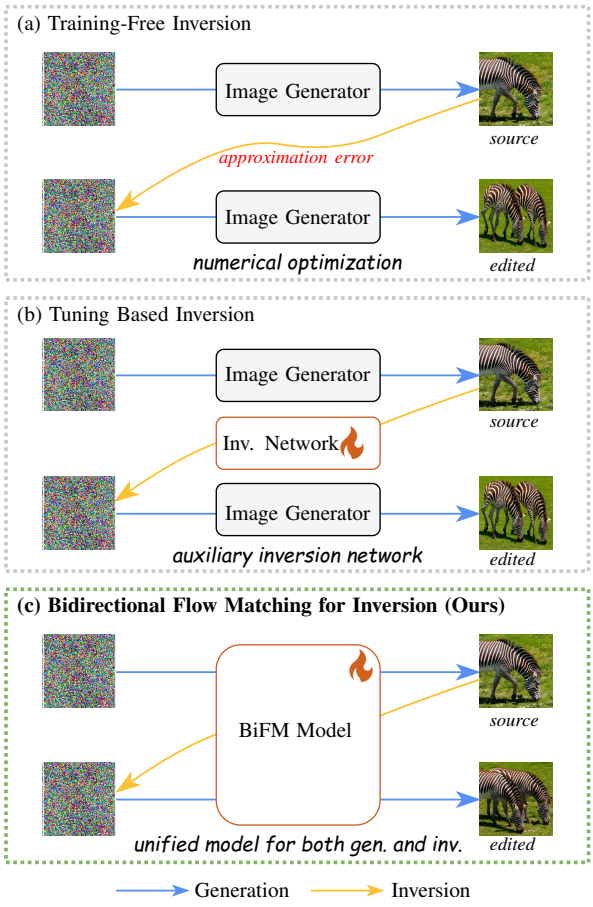


Figure 1. **Inversion-Based Image Editing.** (a) In training-free inversion, the process is approximated by numerically reversing the generation steps, leading to accumulated approximation errors; (b) An auxiliary inversion network is introduced on top of a pretrained generator, improving fidelity but increasing complexity and reducing generalization across architectures. (c) Our method, BiFM, jointly learns generation and inversion within a single flow matching model, enabling consistent few-step inversion and editing.

1. Introduction

Diffusion models [13, 29] and flow matching models [20, 21] achieve strong image generation by learning the data distribution through multi-step sampling. Their generation process can be viewed as solving a learned time-dependent probability flow ordinary differential equation (ODE) [30] starting from random noise. An important application of diffusion models is inversion-based image editing [10, 24, 29, 34], where a source image is mapped back to the intermediate latent space of a generative model and then forwarded again with target prompts. The inversion process enables controllable and semantically faithful edits, but it is naively slow because it doubles the number of inference steps. Consequently, recent research has focused

on few-step editing methods [25, 31, 36], which offer significant advantages in inference speed and enable real-time interactive editing.

However, a key gap remains: the inversion process for few-step diffusion models is intrinsically hard to learn. Few-step models utilize large time-step updates, which amplify the approximation error from local linearization [29] and ODE solvers [23]. An overview of existing inversion-based editing approaches and our proposed bidirectional framework is illustrated in Figure 1: Training-free inversion process inherits approximation errors [14, 32, 40], causing semantic drift or background preservation issues during few-step editing. Tuning based methods [25, 31] learn an inversion process by introducing additional inversion networks and task-specific modules, but incur additional training and computational overhead. Our core motivation arises from this gap: *Can we train a few-step diffusion model that directly learns its own inversion process?*

Motivated by these observations, our research question is how few-step generation and inversion can be jointly learned and how such joint learning can improve generation and editing performance under a few-step sampling budget. To achieve invertibility, bidirectional (invertible) neural networks are often implemented as variants of affine coupling layers [33] that can be explicitly inverted. In contrast, we obtain inversion in a natural way from the ODE viewpoint by integrating the flow matching ODE in both time directions. For few-step generation, instead of learning the entire ODE trajectory, we parameterize the model as the average velocity field of the flow matching ODE over continuous time intervals.

To this end, we introduce **BiFM** (Bidirectional Flow Matching), a flow matching model that directly learns both few-step generation and inversion. We provide a training and sampling pipeline for BiFM that predicts bidirectional average velocities along the flow matching ODE. Following prior work [2, 9, 11], our few-step training uses arbitrary time intervals for supervision. To enable generative training in both time directions, we establish a physically constrained connection between forward (generation) and backward (inversion) average velocity fields by extending the MeanFlow Identity proposed by Geng et al. [11]. To stabilize training, we propose a bidirectional consistency training objective and a lightweight time-interval embedding that can be seamlessly integrated into popular diffusion and flow model backbones, such as SiT and MMDiT.

BiFM offers accurate inversion and image editing and can be either fine-tuned from pretrained flow matching models or trained from scratch. Across a wide range of image editing and generation tasks, BiFM consistently outperforms previous few-step methods. Compared with training-free inversion approaches, BiFM adds minimal training cost while retaining state-of-the-art performance on image editing benchmarks. To justify key design choices, we also conduct ablation studies on one-step image generation.

In summary, our contributions are:

- We propose BiFM, a joint generation and inversion flow matching framework, enabling generation and inversion-based editing under a few-step sampling budget.
- We demonstrate that BiFM can be applied to large pretrained text-to-image diffusion models for efficient fine-tuning on image editing tasks.
- We comprehensively evaluate the performance of BiFM on image editing and various image generation tasks, and provide ablation studies that clarify the impact of core design choices.

2. Related Work

2.1. Inversion based Image Editing

Training-free inversion. DDIM inversion [29] enables editing by reversing the deterministic sampling trajectory. However, its effectiveness relies on local linearity assumptions, which degrade under large step sizes. Subsequent methods [14, 34] introduce solver-specific inversion techniques to mitigate these issues, but still depend on approximate dynamics and lack robustness in few-step regimes.

Tuning few-step/one-step generators. To support interactive editing, recent approaches fine-tune or distill editors from pretrained few-step generators [31], or attach auxiliary inversion modules [25, 36]. While these methods improve efficiency, they often inherit instability from the underlying generators and introduce additional parameters or heuristics for inversion, thus limiting generalization.

2.2. Efficient Diffusion Distillation

Recent approaches to accelerate sampling in diffusion and flow matching models fall into three main categories. First, non-Markovian deterministic sampling methods such as DDIM [29]. Second, progressive distillation [9, 28] that trains compact student models to mimic teacher models. Third, time-interval supervision approaches [11, 35] directly supervise integrated flow matching ODE over continuous time spans, enabling efficient few-step generation. While the third category enables flexible and simplified training recipes, the introduction of time-interval often leads to training instability and typically not applied to inversion, which limits their applicability to inversion-based editing tasks in few sampling steps.

2.3. Invertible Neural Networks

Recent work has explored architectural and training strategies to enable invertibility in generative models. Some approaches impose structural constraints to enforce invertibility [32, 33], while others jointly learn forward and backward mappings through consistency-based distil-

lation [18] or invertible consistency models [31]. For example, iCD [31] demonstrates text-guided editing with around seven steps using consistency distillation and dynamic classifier-free guidance (CFG), highlighting the importance of bidirectional control in text-to-image tasks. While these models show promise for image editing, they have primarily been evaluated on simplified datasets and tasks. The open challenge remains whether such invertible architectures can be effectively applied to complex generative models to reduce inversion error and support high-quality editing in realistic scenarios.

3. Rethink Inversion and Few-Step Diffusion

3.1. Diffusion Model and Flow Matching

Diffusion and flow matching models are generative models that learn a transformation from prior noise distribution $\mathcal{N}(0, I)$ to an unknown complex data distribution $p_{\text{data}}(\mathbf{x})$.

Denoising diffusion models [13, 26, 30] construct this transformation via a forward Markov process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ that gradually adds noise to data, and a learned reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ that denoises it. Training is typically performed by maximizing the log-likelihood of $p_\theta(\mathbf{x}_0)$, which can be simplified to a denoising objective in Eq. (1), following [13] :

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2\right] \tag{1}$$

Flow matching [20, 21] learns a time-dependent velocity field $v_\theta(\mathbf{x}_t, t), t \in [0, 1]$, that drives a continuous flow from noise to data. Given $\mathbf{x}_0 \sim \mathcal{N}(0, I)$ and $\mathbf{x}_1 \sim p_{\text{data}}(\mathbf{x})$, the flow path is defined as a linear interpolation: $\mathbf{x}_t := (1-t)\mathbf{x}_0 + t\mathbf{x}_1$. Flow matching does not rely on a predefined forward process but learns from conditional velocity $v_t|\mathbf{x}_t$ that depends on the coupling between $(\mathbf{x}_0, \mathbf{x}_1)$. An effective objective for training is the conditional flow matching loss [20], as in Eq. (2):

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[||v_\theta(\mathbf{x}_t, t) - v_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)||^2\right] \tag{2}$$

At inference, sampling reduces to solving the learned probability flow ODE $d\mathbf{x}_t/dt = v_\theta(\mathbf{x}_t, t)$, starting from $\mathbf{x}_0$.

Time convention. Below, we use the flow matching time convention $t \in [0, 1]$ (noise $\rightarrow$ data). When relating to diffusion models we treat the discrete diffusion time index as a reparameterization of t.

3.2. Limitations of DDIM-based Inversion

Inversion [5] refers to the process of mapping an image back to its intermediate latent within a pretrained generative model. DDIM [29] inversion is widely used due to its deterministic formulation, which simplifies the sampling process by zeroing σ_t^2, resulting in the following generation step:

$$\tilde{\mathbf{x}}_{t+\Delta t} = \tilde{\mathbf{x}}_t + \delta_t \epsilon_\theta(\mathbf{x}_t, t) \tag{3}$$

where $\tilde{\mathbf{x}}_t := \mathbf{x}_t/\sqrt{\alpha_t}$, $\delta_t := \sqrt{\frac{1-\alpha_{t-1}}{\alpha_{t-1}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}}$. DDIM allows deterministic inversion by reversing Eq. (3):

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t+\Delta t} - \delta_t \epsilon_\theta(\mathbf{x}_t, t) \approx \tilde{\mathbf{x}}_{t+\Delta t} - \delta_t \epsilon_\theta(\mathbf{x}_{t+\Delta t}, t) \tag{4}$$

This training-free inversion mechanism is appealing for image editing tasks. However, in the few-step regime, the discrepancy between consecutive noise predictions, $|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon_\theta(\mathbf{x}_{t+\Delta t}, t)|$, becomes significant due to large step sizes. As illustrated in Figure 2 (c), this leads to poor latent recovery and degraded editing quality, making DDIM inversion unreliable for few-step applications.

3.3. Time-Interval Distillation of Diffusion Models

The need for iterative sampling via ODE/SDE solvers limits efficiency for diffusion and flow models, especially in few-step regimes where speed and fidelity are critical. Similar principle applies to flow matching ODE. By leveraging the physical definitions of average and instantaneous velocity, we define average velocity field $u(\mathbf{x}_t, t, t')$ as the integral of the instantaneous velocity $v(\mathbf{x}_t, t)$ over a time interval $[t, t']$:

$$u(\mathbf{x}_t, t, t') := \frac{1}{t'-t}\int_t^{t'} v(\mathbf{x}_\tau, \tau) d\tau \tag{5}$$

Eq. (5) enables few-step training by supervising the model to match average velocities over time intervals, rather than relying on dense trajectory sampling. However, existing flow matching models typically focus on generation without addressing inversion or editing tasks. In the next section, we introduce BiFM, a bidirectional framework that unifies generation and inversion under a shared velocity field.

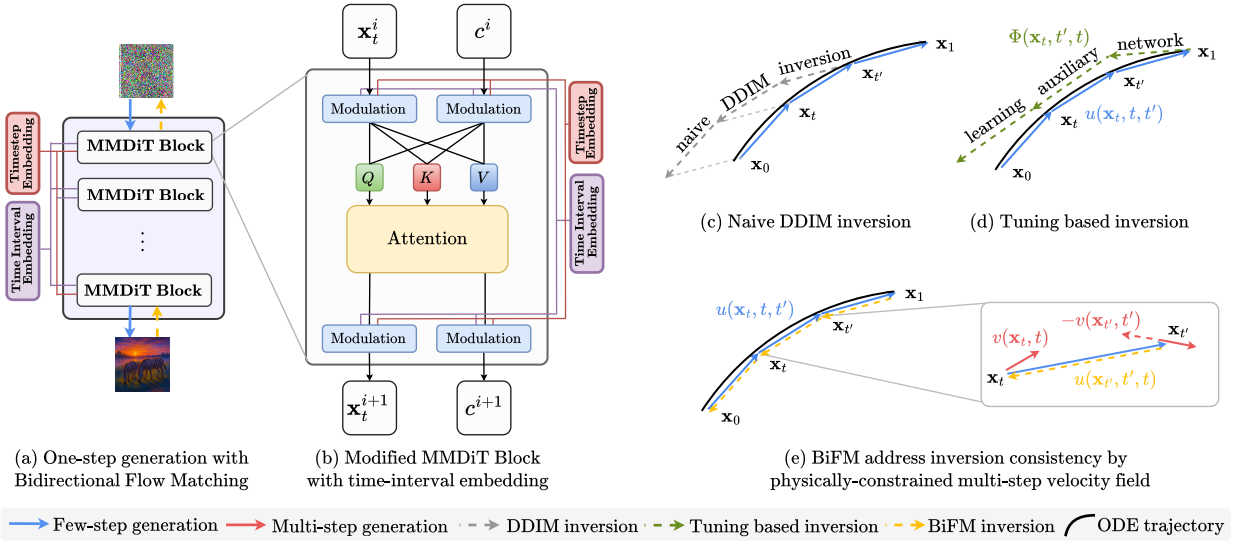
4. Bidirectional Flow Matching (BiFM)

In this section, we introduce the core components of BiFM, our proposed framework for jointly learning few-step generation and inversion. Building on flow matching and time-interval supervision, BiFM extends the MeanFlow Identity to support the modeling of flow matching ODE in both time directions within a single model. Our key insight is that both forward and backward average velocity fields are defined with respect to a shared instantaneous velocity field. This allows BiFM to perform high-fidelity few-step sampling and inversion without relying on DDIM-based inversion or auxiliary modules.

4.1. MeanFlow Identity for One-Step Generator

We start by revisiting the MeanFlow Identity [11] that links the average velocity field $u(\mathbf{x}_t, t, t')$ to the instantaneous velocity $v(\mathbf{x}_t, t)$ in flow matching ODE. Recall the average velocity over a time interval $[t, t']$:

$$u(\mathbf{x}_t, t, t') := \frac{1}{t'-t}\int_t^{t'} v(x_\tau, \tau)\, d\tau. \tag{6}$$



(a) One-step generation with Bidirectional Flow Matching

(b) Modified MMDiT Block with time-interval embedding

(c) Naive DDIM inversion

(d) Tuning based inversion

(e) BiFM address inversion consistency by physically-constrained multi-step velocity field

Figure 2. **Overview of BiFM.** (a) Our one-step generation architecture built upon MMDiT based flow matching model. (b) A single MMDiT block showing how time embedding modulation impact model output. (c) Naive DDIM inversion reuses DDIM update in reverse time, causing departures from original ODE trajectory in few-step regime. (d) Tuning based inversion introduces an auxiliary network $\Phi(\mathbf{x}_{t'}, t', t)$ (e) BiFM inversion (ours) learns a physically constrained bidirectional average velocity field.

Assuming t and t' are independent, differentiating Eq. (6) with respect to t yields Eq. (7), where the total derivative can be expressed using the Jacobian-vector product (JVP) as shown in Eq. (8):

$$u(\mathbf{x}_t, t, t') = v(\mathbf{x}_t, t) + (t' - t) \cdot \frac{d}{dt} u(\mathbf{x}_t, t, t') \quad (7)$$

$$\frac{d}{dt} u(\mathbf{x}_t, t, t') = v(\mathbf{x}_t, t)\partial_{\mathbf{x}_t} u + \partial_t u \quad (8)$$

Together, Eq. (7) and (8) define the MeanFlow Identity that gives expression of target average velocity field u_{tgt} to be computed during training:

$$u_{\text{tgt}} = v(\mathbf{x}_t, t) + (t' - t) \cdot [v(\mathbf{x}_t, t)\partial_{\mathbf{x}_t} u_\theta + \partial_t u_\theta] \quad (9)$$

By introducing Eq. (9), we can train a few-step generation model either from scratch (where $v(\mathbf{x}_t, t)$ is pre-defined schedule like rectified flow [21]) or by fine-tuning (where $v(\mathbf{x}_t, t) := v_\theta(\mathbf{x}_t, t)$ is a pretrained multi-step generator). The training objective $\mathcal{L}_{\text{MF}}$ regresses a parameterized average velocity field u_θ towards u_{tgt}:

$$\mathcal{L}_{\text{MF}} = \mathbb{E}_{t,t',\mathbf{x}} \left[||u_\theta(\mathbf{x}_t, t, t') - \text{sg}(u_{\text{tgt}})||^2 \right] \quad (10)$$

Here $\text{sg}(\cdot)$ denotes stop-gradient operation. Intuitively, u_θ is trained so that taking one average-velocity step from t towards t' approximates integrating the underlying flow matching ODE over $[t, t']$. At convergence, u_θ behaves as a one-step generator consistent with the underlying multi-step dynamics. Figure 2 (e) conceptually illustrates this behavior: multi-step sampling closely follows the underlying flow matching ODE trajectory, while a few-step model aims to approximate the same trajectory using large time intervals through learned average velocities.

4.2. Extend Time Directions for Flow Inversion

Motivation. Most diffusion and flow matching models are trained under a fixed time convention: from noise to data. However, inversion-based editing requires the *opposite* operation: from image to noise. Therefore, DDIM inversion suffers from approximation errors because the model requires $\mathbf{x}_t$ as input to compute $\epsilon_\theta(\mathbf{x}_t, t)$, yet inversion starts from $\mathbf{x}_{t+\Delta t}$, illustrated in Figure 2(c). This mismatch leads to poor inversion in few-step regimes.

Our key observation is that this limitation is largely a consequence of the time convention enforced during training. In the remainder of this subsection, we show how to realize this idea using a bidirectional consistency loss term derived from the MeanFlow Identity.

Bidirectional consistency objective. Although MeanFlow Identity typically assumes $t < t'$ for training and sampling, the identity itself does not depend on this ordering and holds for $t > t'$, allowing us to define inversion using the same formulation. Specifically, given t, t' with $t < t'$, we interpret $u(\mathbf{x}_t, t, t')$ as generation (forward average velocity) and $u(\mathbf{x}_{t'}, t', t)$ as inversion (backward average velocity). Both are defined from the *same* instantaneous velocity field $v(\mathbf{x}_t, t)$, but integrated over opposite time intervals. Apply-

ing Eq. (9) gives:

$$u(\mathbf{x}_t, t, t') = v(\mathbf{x}_t, t) + (t' - t)\big(v(\mathbf{x}_t, t)\partial_{\mathbf{x}_t} u + \partial_t u\big)$$
$$u(\mathbf{x}_{t'}, t', t) = v(\mathbf{x}_{t'}, t') + (t - t')\big(v(\mathbf{x}_{t'}, t')\partial_{\mathbf{x}_{t'}} u + \partial_t u\big)$$

In continuous time, the average velocity for the backward interval $[t', t]$ is the negative of the forward average velocity over $[t, t']$, evaluated at corresponding points along the true trajectory. This is precisely the notion of reversibility we want in a generative model used for inversion-based editing: going forward from (x_t, t) to $(x_{t'}, t')$ and then backward to (x_t, t) should approximately recover the original state, as shown in Figure 2(e).

To encode this reversibility at the level of the learned average velocity, we explicitly encourage the forward and backward predictions to be negatives of each other by introducing a bidirectional consistency loss:

$$\mathcal{L}_{\text{BiFM}} = \mathcal{D}\left(u_\theta(\mathbf{x}_t, t, t'), -u_\theta(\mathbf{x}_{t'}, t', t)\right) \tag{11}$$

where $D(\cdot, \cdot)$ is a distance metric (e.g., a robust ℓ_p norm; see Section 5.5 for choices). This consistency term penalizes the discrepancy between the forward average velocity and backward average velocity. Our final training objective combines the bidirectional consistency term with $\mathcal{L}_{\text{MF}}$:

$$\mathcal{L} = \mathcal{L}_{\text{MF}} + w(t, t') \cdot \mathcal{L}_{\text{BiFM}} \tag{12}$$

Here, $w(t, t')$ is a time-dependent weighting schedule that stabilizes training by gradually strengthening the bidirectional constraint. This formulation allows BiFM to be both trained from scratch and distilled from multi-step models, enabling efficient few-step generation and inversion.

4.3. BiFM Fine-Tuning from Pretrained Models

While most flow matching models adopt a simple velocity field $v_t := \mathbf{x}_1 - \mathbf{x}_0$ during training, BiFM also applies to pretrained diffusion models whose $v_\theta(\mathbf{x}_t, t)$ traces more complex, curved trajectories. Importantly, Eq. (9) does not require explicit access to the instantaneous velocity field, allowing BiFM to be fine-tuned on pretrained models with objective $\mathcal{L}$.

Model Implementation. This flexibility allows BiFM to fine-tune large pretrained flow matching models, such as Stable Diffusion 3, for inversion-based image editing. During fine-tuning, we apply LoRA on model backbone, following settings from Chadebec et al. [4]. As shown in Figure 2(a,b), for encoding time interval embedding input, we augment model backbone with an extra time embedding - we embed t and $(t' - t)$ using standard MLP-based time embeddings, and then add them into a single interval embedding vector. This interval embedding is injected into the network in the same way as the original timestep embedding.

Inference with BiFM. At inference Time, BiFM performs inversion-based editing following Algorithm 1. For trade-off between sampling quality and efficiency, BiFM supports one-step and multi-step sampling by decomposing large time intervals, as shown in Algorithm 2.

In summary, BiFM unifies generation and inversion under a shared velocity field, enabling accurate few-step sampling and editing. Unlike standard diffusion models that rely on numerical ODE/SDE solvers or DDIM-based inversion, it avoids iterative solvers and DDIM approximation errors. Compared to MeanFlow, BiFM extends velocity supervision to both time directions, supporting joint training and fine-tuning for efficient and robust image editing.

Algorithm 1 BiFM: Inversion-Based Editing.

```
# x_1: source image
# p_s: source prompt
# p_t: target prompt
u = model(x_1,1,0,p_s) # inversion
x_0 = x_1 + u
u_edit = model(x_0,0,1,p_t) # generation
x_1_edit = x_0 + u_edit
return x_1_edit
```

Algorithm 2 BiFM: Multi-Step Sampling.

```
# N: number of sample steps
time_steps = linspace(0,1,N)
e = randn(x_shape)
x_0 = e
z = x_0
for i in range(N):
   t_s, t_e = time_steps[i:i+ 2]
   u = model(z,t_s,t_e)
   z = z + (t_e - t_s) * u
x_1 = z
return x_1
```

5. Experiments

We evaluate BiFM as a unified framework for image generation and inversion-based image editing. Our experiments aim to show that: (i) BiFM achieves accurate inversion and reconstruction, (ii) BiFM can be effectively applied to pretrained flow matching models for few-step prompt-guided image editing, and (iii) BiFM enhances image generation sample quality across diverse experimental settings. We also conduct ablation studies on image generation to analyze key design choices and their impact.

5.1. Experiment Setup

Image Editing. We fine-tune Stable Diffusion 3 [8], a state-of-the-art flow matching model, with BiFM and evaluate it on the PIE-Bench [15] benchmark. For image inversion and reconstruction experiment, we use source images and

Settings	Methods	NFE	Background Preservation				CLIP Semantics	
			LPIPS$_{\times 10^3}$ ↓	SSIM$_{\%}$ ↑	PSNR↑	MSE$_{\times 10^4}$ ↓	Whole↑	Edited↑
Multi-Step	NT Inv [24] (CVPR23)	50	60.67	84.11	27.03	35.86	23.61	21.64
	MasaCtrl [3] (ICCV23)	50	106.62	79.67	22.17	86.97	23.96	21.16
	PnP Inv [15] (ICLR24)	50	49.25	84.86	27.22	**32.87**	25.83	22.39
	EditFT [38] (CVPR25)	30	80.55	**91.50**	26.62	40.24	25.74	22.27
	FlowEdit [17] (ICCV25)	28	112.19	83.08	21.96	94.99	25.25	22.58
	DNAEdit [37] (NeurIPS25)	28	112.60	83.69	23.24	67.32	**28.90**	23.66
	BiFM (ours)	50	**47.01**	87.50	**29.89**	42.66	27.42	**24.66**
Few-Step	DDIM [29] (ICLR21)	4	177.96	66.86	18.59	184.69	23.62	21.20
	TurboEdit [36] (ECCV24)	4	76.95	84.63	25.51	58.48	25.49	21.82
	FireFlow [7] (ICML25)	18	-	82.49	23.03	-	26.02	22.81
	InstantEdit [12] (ICCV25)	4	**44.39**	86.44	27.96	**34.94**	26.28	22.82
	ReNoise [10] (NeurIPS25)	4	136.52	76.51	22.59	53.49	24.78	21.37
	BiFM (ours)	4	67.25	**87.29**	**28.92**	54.92	**26.77**	**23.58**
One-Step	SwiftEdit [25] (CVPR25)	1	**91.04**	81.05	23.33	66.58	25.16	21.25
	BiFM (ours)	1	92.30	**85.88**	**28.46**	**58.22**	**26.09**	**22.42**

Table 1. **PIE-Bench Image Editing Performance.** We compare with baselines under different sampling budgets with evaluation metrics from PIE-Bench [15]. **Bold**: best result in the setting, underline: second best.

their corresponding prompts as both input and target. For prompt-guided image editing, we compare BiFM against both training-free inversion methods and few-step editing methods that require fine-tuning or distillation. Evaluation metrics include MSE, PSNR, SSIM, and LPIPS. For semantic alignment, we use CLIP score.

Image Generation. For text-to-image generation on MSCOCO-256 [19], we use an MMDiT backbone from REPA [39] for training from scratch. To validate our design choices, we also assess class-conditional generation on ImageNet-256 [6] (fine-tuning SiT-XL/2) and small-resolution dataset (CIFAR-10 [16]) trained from scratch with a U-Net backbone. We evaluate image generation performance by FID, IS, Precision, and Recall.

Method	MSE$^{\downarrow}_{\times 10^4}$	LPIPS$^{\downarrow}_{\times 10^3}$	SSIM$^{\uparrow}_{\%}$	PSNR↑
DDIM [29] (ICLR21)	224.43	210.84	70.96	17.76
NT Inv [24] (CVPR23)	97.64	102.25	82.33	27.82
PnP Inv [15] (ICLR24)	105.66	95.95	87.20	28.79
RF-Solver [34] (ICML25)	94.80	106.15	86.36	28.26
BiFM (ours)	**87.72**	**89.49**	**88.03**	**30.32**

Table 2. **Image Reconstruction Performance**. We use 50 inversion steps to generate results for all methods. BiFM's learned inversion process greatly reduces reconstruction error.

5.2. Inversion and Reconstruction

To demonstrate that BiFM learns accurate inversion of images, we evaluate image reconstruction results from different inversion methods as well as BiFM itself. We use background preservation metrics from PIE-Bench computed between source images and reconstructed images.

As shown in Table 2, BiFM achieves the best performance on all evaluation metrics, outperforming baselines by a clear margin. In Figure 3, compared to PnP Inversion and RF-Edit, BiFM preserves global scene layout while recovering sharper local details (such as eyes and object geometries), indicating consistent bidirectional flow learning.

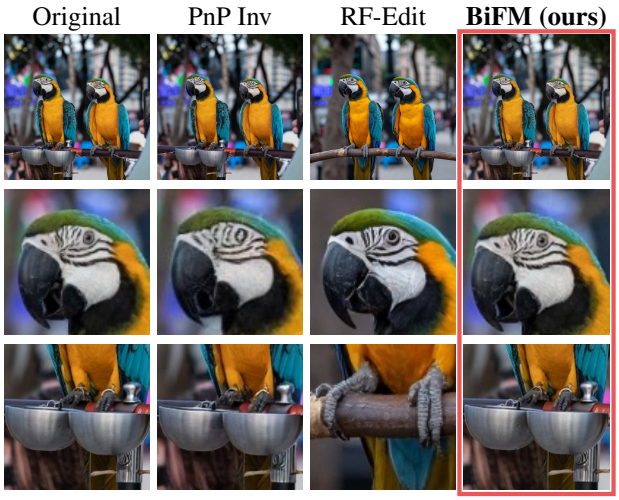


Figure 3. **Inversion and Reconstruction Quality.** From left to right: original input image, PnP Inversion [15], RF-Edit [34], and **BiFM (ours)**. BiFM faithfully reconstructs image details, while RF-Edit exhibits semantic shift and PnP Inv fails to recover fine details in the source image.

5.3. Prompt-Based Image Editing

Model Configuration. We fine-tune a pretrained Stable Diffusion 3 model using BiFM. We use the MagicBrush [41] training dataset for fine-tuning the image editing models. For configuring LoRA, we follow the LoRA hyperparameter settings from Flash Diffusion [4]. Model weights are augmented with an extra time embedding for time-interval representation. The architecture of this additional embedding is identical to the model's original time embedding and is zero-initialized for warm-up. See more model training details in the Appendix.



Figure 4. **Image Editing Visualization.** Given a source image, a source prompt and a target prompt (left illustrates difference between source prompt and target prompt), BiFM generates edits which follow more faithfully the intended concept while better preserving the original layout and fine details than other baselines. For example, BiFM engraves a clear *lion* pattern on the latte art without distorting the background, swaps the *Statue of Liberty's* torch for a flower without geometry distortion, and maintains the *lighthouse* structure.

Baseline Comparison. We compare BiFM with existing image editing methods, including training-free methods and few-step editing methods, under three sampling budgets: multi-step, few-step, and one-step. Table 1 summarizes the results on PIE-Bench across background preservation and CLIP-based semantic metrics.

Multi-Step. In the multi-step setting (50 NFEs), BiFM achieves the best overall balance between reconstruction fidelity and semantic alignment. Compared with training-free baselines such as MasaCtrl [3] and PnP Inv [15], BiFM improves both background preservation metrics and CLIP semantics, indicating that it better preserves background content while producing more faithful edits.

Few-Step & One-Step. Under the 4-step regime, BiFM attains LPIPS 67.25, SSIM 87.29, and PSNR 28.92, outperforming both training-free inversion [10, 29] and auxiliary network methods [36], while remaining competitive on MSE. This supports our claim that explicitly learning forward/backward average velocities within a single network stabilizes few-step inversion-editing. Compared to SwiftEdit [25], BiFM trades a slightly higher LPIPS (92.30 $vs.$ 91.04) for materially better SSIM, PSNR, MSE, and CLIP semantics. This suggests BiFM favors structural/semantic preservation in the extreme one-step regime.

Editing Visualization. As shown in Figure 4, BiFM produces edits that better satisfy the target prompts while more faithfully preserving object structure and background details than baselines.

5.4. Image Generation

We conduct image generation experiments on both small and large resolution datasets. We explore two training settings for BiFM: training from scratch and fine-tuning.

Text-to-Image Generation: For text-to-image generation on MSCOCO-256, we train a vanilla MMDiT following the [39] configuration and evaluate FID on 40k samples (equal to the validation set size) using REPA configurations without representation alignment. Table 4 shows evaluation results on FID. With MMDiT, BiFM reduces FID from 6.05 (vanilla) and 4.73 (REPA) to 4.57, validat-

Time Input	FID↓
$(t, t', \text{direc.})$	69.01
$t' - t$	60.86
(t, t')	59.37
$(t, t' - t)$	**55.22**

(a) **Timestep embedding.** The model is conditioned on specific time parameterization designs.

Time Sampler	FID↓
uniform(0, 1)	63.25
lognorm(–0.4, 1.2)	59.16
lognorm(–0.2, 1.0)	58.57
lognorm(–0.4, 1.0)	**55.22**

(b) **Time sampler.** t and t' are sampled from the specific distributions during training.

Weighting Func.	FID↓
linear	67.37
sin	62.45
log	57.93
warm up	**55.22**

(c) **Weighting function**. A warm-up schedule on the consistency term yields the best performance.

Loss Norm	FID↓
0.0	72.84
2.0	59.66
0.5	58.74
1.0	**55.22**

(d) **Loss norm metrics.** p=0 is squared L2 loss. p=0.5 is Pseudo-Huber loss.

Table 3. **Ablation Study on 1-NFE Image Generation.** FID computed from 50K samples is reported. Defaults are shown in bottom row.

ing that our bidirectional average-velocity training complements representation-alignment style improvements.

Method	Model	FID↓
U-Net [27]	Diffusion	7.32
U-ViT-S/2 [1]	Diffusion	5.95
U-ViT-S/2 (Deep) [1]	Diffusion	5.48
MMDiT	Flow Matching	6.05
MMDiT+REPA	Flow Matching	<u>4.73</u>
MMDiT+MeanFlow	Flow Matching	5.02
MMDiT+ BiFM (ours)	Flow Matching	**4.57**

Table 4. **MSCOCO-256 Text-to-Image Generation.** FID computed from 40K samples is reported.

Setting	Model	FID↓	NFE
Multi-Step	DDIM [29]	4.67	50
	Flow Matching [20]	<u>2.63</u>	50
	BiFM (ours)	**2.17**	50
One-Step	Rectified Flow [21]	4.85	1
	sCT [22]	<u>2.85</u>	1
	MeanFlow [11]	2.92	1
	BiFM (ours)	**2.75**	1

Table 5. **Unconditional CIFAR-10 Results.** We include both multi-step and one-step FID.

Few-Step Generation. For the CIFAR-10 setting, we use the Flow Matching [20] configuration. The model is a U-Net backbone trained from scratch for 500 epochs, Table 5 shows quantitative results. On CIFAR-10, BiFM improves both multi-step and one-step FID. With 50 NFE, BiFM reaches FID 2.17, improving over Flow Matching. With 1 NFE, BiFM sets the best FID on CIFAR-10 (2.75 vs. 2.85 for sCT and 2.92 for MeanFlow).

ImageNet-256 training from scratch. In Table 6, when training vanilla SiT variants from scratch, BiFM consistently lowers FID across model scales (e.g., SiT-XL/2: 17.2 → 15.5), indicating benefits beyond fine-tuning.

5.5. Ablation Studies

We ablate key design choices in 1-NFE ImageNet-256 training by varying one component at a time while keeping the architecture and training budget fixed; Table 3 summarizes the effect on FID and marks our default settings.

Model	Epochs	FID↓
SiT-B/2	80	33.0
SiT-B/2+ **BiFM**	80	**27.8**
SiT-L/2	80	18.8
SiT-L/2+ **BiFM**	80	**16.4**
SiT-XL/2	80	17.2
SiT-XL/2+ **BiFM**	80	**15.5**

Table 6. **ImageNet Performance across Model Size.** Under multi-step sampling, BiFM training also consistently improves performance across model sizes.

Time-interval conditioning. Conditioning on $(t, t' - t)$ yields the best FID (55.22) versus (t, t') (59.37), $(t' - t)$ alone (60.86), or adding a discrete direction flag (69.01). Interpreting $(t' - t)$ as an explicit interval length helps the network to model the integrated flow over variable spans—directly matching our average-velocity target.

Sampling of (t, t'). Among uniform and log-normal samplers, the best-performing setting skews toward shorter intervals while retaining coverage of longer hops. Practically, biasing samples to smaller $|t' - t|$ stabilizes training early.

Consistency weighting $w(t, t')$. We found a warm-up profile avoids over-regularizing at initialization and encourages bidirectional agreement as predictions sharpen.

Loss norm p. Transitioning from pure L2 $(p = 0)$ to a robust loss (e.g., p≈0.5) improves stability and FID by soft-clipping large residuals from difficult intervals.

6. Conclusion

In this work, we introduce BiFM, a novel framework for jointly learning few-step image generation and inversion within a single model. BiFM extends flow matching continuous time-interval supervision to both time directions to deliver accurate few-step editing. We validate its effectiveness on inversion-based image editing and generation tasks, where BiFM consistently outperforms baselines. We also conduct ablation studies on 1-NFE image generation to justify our design choices.

Acknowledgments

This research is funded in part by an ARC (Australian Research Council) Discovery Grant of DP220100800. Prof. Hongdong Li holds concurrent appointments as a Full Professor with the ANU and as an Amazon Scholar with Amazon (part time). This paper describes work performed at ANU and is not associated with Amazon.

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A ViT backbone for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2023. 8

[2] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. 2

[3] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023. 6, 7

[4] Clement Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025. 5, 6, 11

[5] Yinan Chen, Jiangning Zhang, Yali Bi, Xiaobin Hu, Teng Hu, Zhucun Xue, Ran Yi, Yong Liu, and Ying Tai. Image inversion: A survey from gans to diffusion and beyond, 2025. 3

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[7] Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. Fireflow: Fast inversion of rectified flow for image semantic editing. *arXiv preprint arXiv:2412.07517*, 2024. 6

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 5, 11

[9] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2

[10] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024. 1, 6, 7

[11] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 2, 3, 8

[12] Yiming Gong, Zhen Zhu, and Minjia Zhang. Instantedit: Text-guided few-step image editing with piecewise rectified flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16808–16817, 2025. 6

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3, 11

[14] Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7069–7078, 2024. 2

[15] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024. 5, 6, 7

[16] Alex Krizhevsky et al. Learning multiple layers of features from tiny images, 2009. 6

[17] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19721–19730, 2025. 6

[18] Liangchen Li and Jiajun He. Bidirectional consistency models. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. 3

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[20] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 3, 8, 12

[21] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 1, 3, 4, 8

[22] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024. 8

[23] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022. 2

[24] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 1, 6

[25] Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *Proceedings of the*

Computer Vision and Pattern Recognition Conference, pages 21492–21501, 2025. 2, 6, 7

[26] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 3

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 8

[28] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2, 3, 6, 7, 8

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 3

[31] Nikita Starodubcev, Mikhail Khoroshikh, Artem Babenko, and Dmitry Baranchuk. Invertible consistency distillation for text-guided image editing in around 7 steps. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 3

[32] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023. 2

[33] Fangyikang Wang, Hubery Yin, Yue-Jiang Dong, Huminhao Zhu, Hanbin Zhao, Hui Qian, Chen Li, et al. Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *Advances in Neural Information Processing Systems*, 37:46118–46159, 2024. 2

[34] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. In *Forty-second International Conference on Machine Learning*, 2025. 1, 2, 6

[35] Zidong Wang, Yiyuan Zhang, Xiaoyu Yue, Xiangyu Yue, Yangguang Li, Wanli Ouyang, and Lei Bai. Transition models: Rethinking the generative learning objective, 2025. 2

[36] Zongze Wu, Nicholas Kolkin, Jonathan Brandt, Richard Zhang, and Eli Shechtman. Turboedit: Instant text-based image editing, 2024. 2, 6, 7

[37] Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. Dnaedit: Direct noise alignment for text-guided rectified flow editing. *arXiv preprint arXiv:2506.01430*, 2025. 6

[38] Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, Charles Ling, and Boyu Wang. Unveil inversion and invariance in flow transformer for versatile image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28479–28489, 2025. 6

[39] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025. 6, 7

[40] Guoqiang Zhang, J. P. Lewis, and W. Bastiaan Kleijn. Exact diffusion inversion via bi-directional integration approximation, 2023. 2

[41] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 6, 11