Taming Rectified Flow for Inversion and Editing

Jiangshan Wang[1,2*], Junfu Pu[2*†], Zhongang Qi[2‡], Jiayi Guo[1], Yue Ma[3], Nisha Huang[1], Yuxin Chen[2], Xiu Li[1‡], Ying Shan[2]

[1]Tsinghua University [2]ARC Lab, Tencent PCG [3]HKUST

https://github.com/wangjiangshan0725/RF-Solver-Edit

Figure 1. We propose RF-Solver to solve the rectified flow ODE with less error, thus enhancing both sampling quality and inversion-reconstruction accuracy for rectified-flow-based generative models. Furthermore, we propose RF-Edit to leverage the RF-Solver for image and video editing tasks. Our methods achieve impressive performance on various tasks, including text-to-image generation, image & video inversion, and image & video editing.

Abstract

Rectified-flow-based diffusion transformers, such as FLUX and OpenSora, have demonstrated exceptional performance in the field of image and video generation. Despite their robust generative capabilities, these models often suffer from inaccurate inversion, which could further limit their effectiveness in downstream tasks such as image and video editing. To address this issue, we propose RF-Solver, a novel training-free sampler that enhances inversion precision by reducing errors in the process of solving rectified flow ODEs. Specifically, we derive the exact formulation of the rectified flow ODE and perform a high-order Taylor expansion to estimate its nonlinear components, significantly decreasing the approximation error at each timestep. Building upon RF-Solver, we further design RF-Edit, which comprises specialized sub-modules for image and video editing. By sharing self-attention layer features during the editing process, RF-Edit effectively preserves the structural information of the source image or video while achieving high-quality editing results. Our approach is compatible with any pre-trained rectified-flow-based models for image and video tasks, requiring no additional training or optimiza-

*Equal contribution.
†Project lead.
‡Corresponding authors.

tion. Extensive experiments on text-to-image generation, image & video inversion, and image & video editing demonstrate the robust performance and adaptability of our methods. Code is available at this URL.

1. Introduction

Recently, rectified-flow-based generation methods [31] have demonstrated remarkable performance in generating high-quality images and videos. Compared with traditional approaches such as Stable Diffusion [19, 42], their impressive capabilities stem primarily from two aspects. First, rectified flow utilizes a straightforward Ordinary Differential Equation (ODE), which constructs a continuous straight-line motion system to produce the desired data distribution. This simplified training and inference paradigm enables the model to more effectively learn the underlying distribution of real image data. Second, most of the latest rectified flow methods employ the Diffusion Transformer (DiT) [38, 59] architecture as the backbone, which has shown superior performance compared to traditional U-Net [43] architectures. As a result, image generation models like FLUX [1] and video generation models like OpenSora [2], both based on rectified flow and DiT, have respectively become one of the state-of-the-art (SOTA) models in Text-to-Image (T2I) and Text-to-Video (T2V) generation.

Apart from fundamental T2I and T2V tasks, other downstream tasks such as reconstruction [16, 35, 46] and editing [17] have been attracting growing interest. These tasks typically depend on performing inversion on the source image/video. The inversion process yields the corresponding representation in the noise space, which is followed by denoising with various conditions provided by users. Given the robust generative capabilities of rectified-flow-based models, they are expected to exhibit superior performance on these downstream tasks. However, their performance remains unsatisfactory compared with traditional methods (based on DDPM and UNets), and research in this area still lags behind.

Delving into this problem, we identify that the primary challenge lies in the significant errors during the inversion process of the rectified flow, which fail to accurately reconstruct the original images or videos (Task 1 and Task 2 in Fig. 1). This limitation further constrains its performance on other downstream tasks such as editing. Although [45] attempted to address this challenge through dynamic optimal control, its application is restricted to simple image editing scenarios such as stylization and facial editing, which typically involve simple content and uniform backgrounds. In contrast, both the image contents and editing requirements (such as addition, replacement, and global editing) are considerably more complex in the real world. Moreover, video editing requires highly consistent temporal modeling, presenting even greater challenges for editing algorithms. To the best of our knowledge, there are still no existing methods that effectively tackle these issues.

Instead of focusing on designing a specific inversion method, we aim to address the above problem from a more general and fundamental perspective: the sampler. This is because the essence of the inversion and generation process for rectified flow is to employ a sampler that estimates the solution of rectified flow ODE. Consequently, the primary source of inaccurate inversion lies in the approximation error in the solution, which accumulates at each timestep. Intuitively, if the ODE is solved more accurately, the accuracy of inversion can be enhanced subsequently.

Based on this insight, we propose the **RF-Solver**. Specifically, we note that the ODE formulation for rectified flow can be solved directly using the variation of constants method, yielding an exact formulation of the solutions. For the nonlinear component of this solution (i.e., the integral of the neural network), we utilize Taylor expansion for estimation. By employing higher-order Taylor expansion, the ODE can be solved with reduced error, thereby improving the performance of rectified flow models. RF-Solver is a generic sampler that can be seamlessly integrated into any rectified flow model without the need for training or optimization. Experimental results demonstrate that RF-Solver not only significantly enhances the accuracy of inversion and reconstruction, but also improves performance on fundamental tasks such as T2I generation.

Building upon this, we propose **RF-Edit** to apply RF-Solver in editing tasks. Real-world image and video editing require the model to make precise modifications to a source image/video while maintaining its overall structure unchanged. This makes editing a more challenging task compared to reconstruction. In this scenario, it is inadequate to solely rely on the inverted noises as prior knowledge for editing, which could lead to edited results being excessively influenced by the target prompt, resulting in a completely different output compared with source image/video. Addressing this problem, we store the V (value) feature in the self-attention layers at several timesteps during inversion. In the process of denoising, these features are used to replace the corresponding features. Practically, we design two specific sub-modules for RF-Edit, respectively leveraging FLUX [1] and OpenSora [2] as the backbone for image and video editing. With the effective design of RF-Edit, it demonstrates superior performance in both image and video domains, outperforming various SOTA methods.

Our core contributions are summarized as follows:

- We propose RF-Solver, a training-free sampler that significantly reduces errors in the inversion and reconstruction processes of the rectified-flow model.
- We present RF-Edit, which leverages RF-Solver for image and video editing, effectively preserving the structural

integrity of the source image/video while achieving high-quality results.
- Extensive experiments on images and videos demonstrate the efficacy of our methods, showcasing superior performance in both inversion and high-quality editing compared to various existing baselines.

2. Related Work

2.1. Inversion

Inversion maps the real visual data, *i.e*. image and video, to a representation in the noise space [14, 23, 34, 35, 44, 50], which is the reverse process of generation. This representation captures the essential features and structures of the original data while allowing for flexibility in manipulation for various editing applications. Numerous previous inversion approaches have been elaborately designed for diffusion models to achieve remarkable performance. The representative method, DDIM inversion [46, 47], adds predicted noise recursively at each forward step, outputting the final state as structured noise. To mitigate the discretization error in DDIM inversion, some efforts have been explored from the perspective of optimizing null prompt embeddings [35] and latent variables [44]. Negative prompt inversion [34] accelerates the inversion process at the cost of fidelity. Despite the success of inversion in diffusion models, the exploration of inversion in SOTA rectified flow models like FLUX and OpenSora is limited. RF-prior [57] uses the score distillation to invert the image while it requires a number of optimizing steps. More recently, [45] introduces an additional vector field that is conditioned on the source image to improve the inversion. However, its performance improvement mainly stems from the information leakage of the source image during the inversion and reconstruction process. The error from the original vector field of rectified flow still persists, which would limit the performance of such method on various downstream tasks.

2.2. Image and Video Editing

Training-free methods for image and video editing [21, 48] have gained increasing popularity due to their efficiency and effectiveness. These methods usually invert the source image/video into Gaussian noises and then denoises it conditioned on the target prompt. Existing image editing methods focus on prompt refinement [41, 52], attention-sharing mechanism [7, 17, 37, 49], mask guidance [5, 11, 20, 28], and noise initialization [6, 58]. Video editing introduces additional complexities due to the need for maintaining temporal consistency, making it a more challenging task. Existing video editing methods focus on attention injection [30, 39, 53], motion guidance [10, 15, 51, 56], latent manipulation [9, 25, 55, 61], and canonical representation [8, 26, 27, 36]. To date, the editing performance of rectified-flow-based diffusion transformers has remained largely under-explored. Although [45] employs FLUX for image editing, its performance is limited to simple tasks such as stylization and face editing, and it struggles to effectively maintain the structural information of source images. Moreover, currently there is no research exploring the video editing capabilities of rectified-flow-based models.

3. Method

In this section, we present our methods in detail. First, we introduce the proposed RF-Solver, which significantly enhances the precision of inversion and reconstruction. Subsequently, we introduce RF-Edit, an extension of RF-Solver designed to enable high-quality image and video editing.

3.1. Preliminaries

Rectified Flow [32] facilitates the transport between the Gaussian Noises distributions π_0 and real data distribution π_1 along a straight path. This is achieved by learning a forward-simulating system given by $\frac{d\boldsymbol{Z}_t}{dt} = v(\boldsymbol{Z}_t, t), t \in [0, 1]$ which maps $\boldsymbol{Z}_0 \in \pi_0$ to $\boldsymbol{Z}_1 \in \pi_1$. In practice, the velocity field v is parameterized by a neural network $\boldsymbol{v}_\theta(\boldsymbol{Z}_t, t)$. During training, given $\boldsymbol{X}_0 \in \pi_0$ and $\boldsymbol{X}_1 \in \pi_1$, the forward process of rectified flow is implemented through a simple linear combination:

$$\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0, t \in [0, 1]. \tag{1}$$

The rectified flow ODE can be derived by differentiating Eq. (1) with respect to t, as follows:

$$\frac{d\boldsymbol{X}_t}{dt} = \boldsymbol{X}_1 - \boldsymbol{X}_0, t \in [0, 1]. \tag{2}$$

Here, $\boldsymbol{X}_1 - \boldsymbol{X}_0$ serves as the ground truth, and the network is optimized through mean square error:

$$\min_\theta \int_0^1 \mathbb{E}\left[\left\|(\boldsymbol{X}_1 - \boldsymbol{X}_0) - \boldsymbol{v}_\theta\left(\boldsymbol{X}_t, t\right)\right\|^2\right] dt. \tag{3}$$

During the sampling process, the ODE is discretized and solved using Euler method. Specifically, the rectified flow model starts with Gaussian noises $\boldsymbol{Z}_{t_N} \in \mathcal{N}(0, \boldsymbol{I})$. Given a series of discrete N timesteps $t = \{t_N, ..., t_0\}$, the model iteratively predicts $\boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i), i \in \{N, \cdots, 1\}$ and then takes a step forward until generating the images $\boldsymbol{Z}_{t_0}$, with the following recurrence relation:

$$\boldsymbol{Z}_{t_{i-1}} = \boldsymbol{Z}_{t_i} + (t_{i-1} - t_i)\boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i). \tag{4}$$

The definition and training process of Rectified Flow is designed to establish a nearly linear transition trajectory between two distributions, enabling efficient generation with significantly fewer steps compared to DDPMs [19].

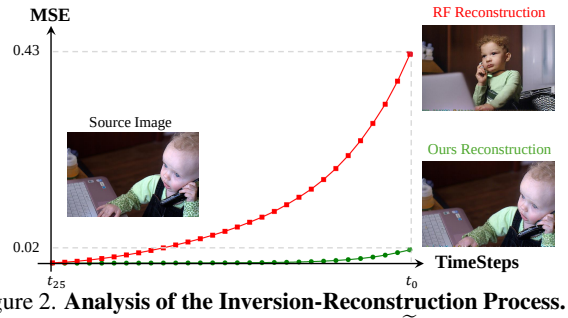


Figure 2. **Analysis of the Inversion-Reconstruction Process.** Inversion begins with the source image latent $\widetilde{\boldsymbol{Z}}_{t_0}$ as the input and progressively add noise, obtaining $\widetilde{\boldsymbol{Z}}_{t_N} \in \mathcal{N}(0, \boldsymbol{I})$. $\widetilde{\boldsymbol{Z}}_{t_N}$ is then denoised for N timesteps to obtain the reconstruction $\boldsymbol{Z}_{t_0}$. During this process, we store the latent $\widetilde{\boldsymbol{Z}}_{t_i}$ and $\boldsymbol{Z}_{t_i}$ at each time step and calculate the mean squared error between them.

3.2. RF-Solver

The vanilla rectified flow (RF) sampler demonstrates strong performance in image and video generation. However, when applied to inversion and reconstruction tasks, we observe significant error accumulation at each timestep. This results in reconstructions that diverge notably from the original image (see Fig. 2), further limiting the performance of RF-based models in various downstream tasks, such as image and video editing. Delving into this problem, we notice that the inversion and reconstruction processes in rectified flow rely on estimating an approximate solution of the rectified flow ODE at each timestep (see Eq. (4)). Obtaining more precise solutions for the ODE would effectively mitigate these errors, leading to improved reconstruction quality. Based on this analysis, we start by carefully examining the differential form of the Rectified flow: $\frac{d\boldsymbol{Z}_t}{dt} = \boldsymbol{v}_\theta(\boldsymbol{Z}_t, t)$. This ODE is discretized in the sampling process. Given the initial value $\boldsymbol{Z}_{t_i}$, the ODE can be exactly formulated using the *variant of constant* method:

$$\boldsymbol{Z}_{t_{i-1}} = \boldsymbol{Z}_{t_i} + \int_{t_i}^{t_{i-1}} \boldsymbol{v}_\theta(\boldsymbol{Z}_\tau, \tau) d\tau. \tag{5}$$

In the above formula, $\boldsymbol{v}_\theta(\boldsymbol{Z}_\tau, \tau)$ is the non-linear component parameterized by the complex neural network, which is difficult to approximate directly. As an alternative, we employ the Taylor expansion at t_i to approximate this term:

$$\boldsymbol{v}_\theta(\boldsymbol{Z}_\tau, \tau) = \sum_{k=0}^{n-1} \frac{(\tau - t_i)^k}{k!} \boldsymbol{v}_\theta^{(k)}(\boldsymbol{Z}_{t_i}, t_i) + \mathcal{O}\big((\tau - t_i)^n\big), \tag{6}$$

where $\boldsymbol{v}_\theta^{(k)}(\boldsymbol{Z}_{t_i}, t_i) = \frac{\mathrm{d}^k \boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i)}{\mathrm{d}t^k}$, denoting the k-order derivative of $\boldsymbol{v}_\theta$ and $\mathcal{O}$ denotes higher-order infinitesimals. Substituting Eq. (6) into the integral term yields:

$$\int_{t_i}^{t_{i-1}} \boldsymbol{v}_\theta(\boldsymbol{Z}_\tau, \tau)\, d\tau = \sum_{k=0}^{n-1} \boldsymbol{v}_\theta^{(k)}(\boldsymbol{Z}_{t_i}, t_i) \int_{t_i}^{t_{i-1}} \frac{(\tau - t_i)^k}{k!} d\tau + \mathcal{O}\big((\tau - t_i)^n\big). \tag{7}$$

Algorithm 1 Sampling process of RF-Solver

Input:

$\boldsymbol{v}_\theta$ ▷ *Velocity function*
$t = [t_N, \dots, t_0]$ ▷ *Time steps*
$\boldsymbol{Z}_{t_N} \sim \mathcal{N}(0, I)$ ▷ *Initial Gaussian Noise*

For $i = N$ **to** 1 **do**

$\Delta t_i \leftarrow \frac{1}{2}(t_{i-1} - t_i)$
$\hat{\boldsymbol{v}}_{t_i} \leftarrow \boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i)$
$\boldsymbol{Z}_{t_i + \Delta t_i} \leftarrow \boldsymbol{Z}_{t_i} + \Delta t_i \hat{\boldsymbol{v}}_{t_i}$
$\hat{\boldsymbol{v}}_{t_i + \Delta t_i} \leftarrow \boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i + \Delta t_i}, t_i + \Delta t_i)$
$\boldsymbol{v}_{t_i}^{(1)} \leftarrow (\hat{\boldsymbol{v}}_{t_i} - \hat{\boldsymbol{v}}_{t_i + \Delta t_i})/\Delta t_i$ ▷ *Calculating the Derivatives*
$\boldsymbol{Z}_{t_{i-1}} \leftarrow \boldsymbol{Z}_{t_i} + (t_{i-1} - t_i)\hat{\boldsymbol{v}}_{t_i} + \frac{1}{2}(t_{i-1} - t_i)^2 \boldsymbol{v}_{t_i}^{(1)}$

Output: $\boldsymbol{Z}_0$

Through the above process, the network prediction term and its higher-order derivatives are separated from the integral. Then we notice that the remaining portion in the integral can be computed analytically:

$$\int_{t_i}^{t_{i-1}} \frac{(\tau - t_i)^k}{k!} d\tau = \left[\frac{(\tau - t_i)^{k+1}}{(k+1)!} \right]_{t_i}^{t_{i-1}} = \frac{(t_{i-1} - t_i)^{k+1}}{(k+1)!}. \tag{8}$$

Substituting Eq. (8) and Eq. (7) into Eq. (5), we derive the n-th order solution of Rectified flow ODE:

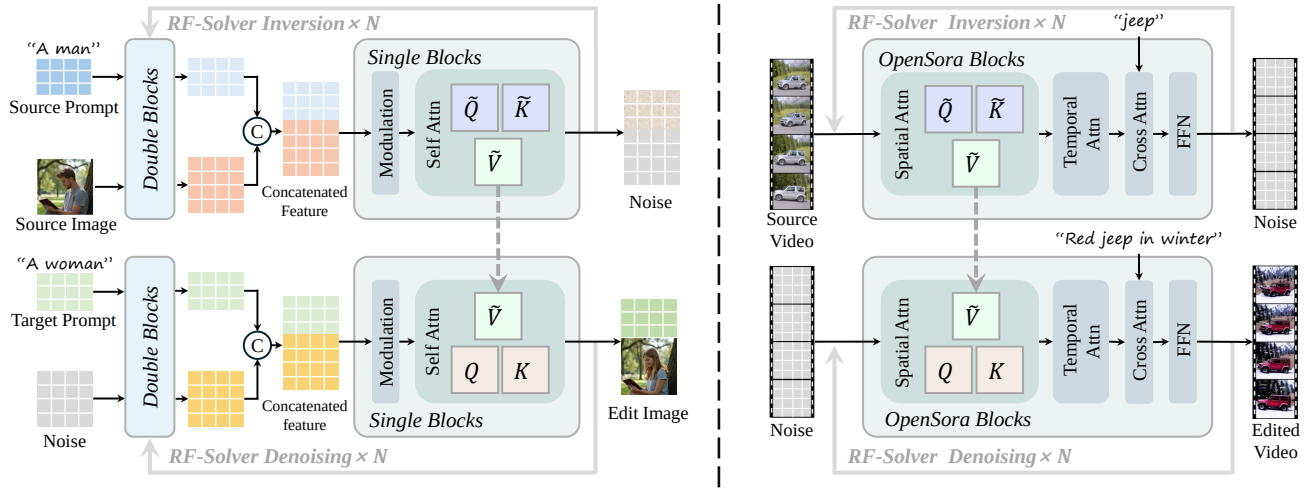
$$\boldsymbol{Z}_{t_{i-1}} = \boldsymbol{Z}_{t_i} + \sum_{k=0}^{n-1} \frac{(t_{i-1} - t_i)^{k+1}}{(k+1)!} \boldsymbol{v}_\theta^{(k)}(\boldsymbol{Z}_{t_i}, t_i) + \mathcal{O}\big(h_i^{n+1}\big), \tag{9}$$

where $h_i := t_{i-1} - t_i$. Eq. (9) indicates that to estimate $\boldsymbol{Z}_{t_{i-1}}$, we need to obtain the k-th order derivatives $\{\boldsymbol{v}_\theta^{(k)}(\boldsymbol{Z}_{t_i}, t_i)\}$ for $k \in \{0, \cdots, n-1\}$. When $n = 1$, the formula reduces to the standard rectified flow (Eq. (4)). In our experiments, we find that setting $n = 2$ effectively mitigates the errors, yielding the following formula:

$$\boldsymbol{Z}_{t_{i-1}} = \boldsymbol{Z}_{t_i} + (t_{i-1} - t_i)\boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i) + \frac{1}{2}(t_{i-1} - t_i)^2 \boldsymbol{v}_\theta^{(1)}(\boldsymbol{Z}_{t_i}, t_i). \tag{10}$$

Note that $\boldsymbol{v}_\theta^{(1)}$ is the first-order derivative of the network prediction term $\boldsymbol{v}_\theta$, which cannot be analytically derived due to the complex architecture of the neural network. To estimate this term, we first obtain the network prediction $\hat{\boldsymbol{v}}_{t_i}$ at the timestep t_i, *i.e.*, $\hat{\boldsymbol{v}}_{t_i} = \boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i}, t_i)$. Then we step forward a small timestep $\Delta t = \frac{1}{2}(t_{i-1} - t_i)$, and update the latents to obtain $\boldsymbol{Z}_{t_i + \Delta t} = \boldsymbol{Z}_{t_i} + \Delta t \cdot \hat{\boldsymbol{v}}_{t_i}$. Next, we calculate an additional prediction of the network at the timestep $t_i + \Delta t$, *i.e.*, $\hat{\boldsymbol{v}}_{t_i + \Delta t} = \boldsymbol{v}_\theta(\boldsymbol{Z}_{t_i + \Delta t}, t_i + \Delta t)$. With $\hat{\boldsymbol{v}}_{t_i}$ and $\hat{\boldsymbol{v}}_{t_i + \Delta t}$, the first-order derivative of $\boldsymbol{v}_\theta$ at the timestep t_i can be estimated as:

$$\boldsymbol{v}_\theta^{(1)}(\boldsymbol{Z}_{t_i}, t_i) = \frac{\hat{\boldsymbol{v}}_{t_i + \Delta t} - \hat{\boldsymbol{v}}_{t_i}}{\Delta t}. \tag{11}$$



(a). RF-Editing on Image (a). RF-Editing on Video

Figure 3. **RF-Edit Pipelines.** The RF-Edit framework comprises two sub-modules, respectively applied to FLUX [1] for image editing and OpenSora [2] for video editing. For image editing, we share the feature of $\widetilde{\mathcal{V}}$ within the single block self-attention of the FLUX [1] backbone. For video editing, we share the feature of $\widetilde{\mathcal{V}}$ in the spatial attention of the OpenSora [2] backbone.

Substituting Eq. (11) into Eq. (10) results in the practical implementation of the RF-Solver algorithm. The complete sampling process for RF-Solver is presented in Algorithm 1.

Besides sampling, inversion seeks to map data back into noise, which reverses the sampling process. Following previous methods for DDIM inversion [13, 46], the ODE process can be directly reversed in the limit of small steps. Based on this assumption, the inversion process of RF-Solver (Eq. (10)) can be directly transformed as:

$$\begin{aligned}\widetilde{\boldsymbol{Z}}_{t_{i+1}} = \widetilde{\boldsymbol{Z}}_{t_i} &+ (t_{i+1} - t_i)\boldsymbol{v}_\theta(\widetilde{\boldsymbol{Z}}_{t_i}, t_i) \\ &+ \frac{1}{2}(t_{i+1} - t_i)^2 \boldsymbol{v}_\theta^{(1)}(\widetilde{\boldsymbol{Z}}_{t_i}, t_i),\end{aligned} \tag{12}$$

where $\widetilde{\boldsymbol{Z}}_{t_i}$ and $\widetilde{\boldsymbol{Z}}_{t_{i+1}}$ denotes the latents during inversion. Through this high order expansion, the error of the ODE solution in each timestep is reduced from $\mathcal{O}\big((h_i)^2\big)$ to $\mathcal{O}\big((h_i)^3\big)$, leading to improved performance, particularly in inversion and reconstruction (see Fig. 2). Beyond inversion and reconstruction, RF-Solver can also be applied to any RF-based model (such as FLUX [1] and OpenSora [2]) for other tasks such as sampling and editing, enhancing performance without requiring additional training.

3.3. RF-Edit

Incorporating higher-order terms enables RF-Solver to significantly reduce errors in the ODE-solving process, improving both sampling quality and inversion accuracy. Furthermore, we extend the application of RF-Solver to real-world image and video editing tasks, which are more challenging than reconstruction. In these scenarios, maintaining the content and structure of the original image is crucial. For instance, when adding new objects to a source image, other unrelated regions should remain unaffected by the editing process. However, directly applying RF-Solver during the inversion and denoising stages may cause the model to be overly influenced by the target prompt, leading to unintended modifications in other parts of the image or video (e.g., altering the unrelated which is not mentioned in the editing prompt). This issue is common across various existing editing methods [17, 45, 49].

To address this problem, we propose RF-Edit, which incorporates features from the inversion process into the denoising procedure. Specifically, during the last n steps of inversion, we extract and store the Value feature $\{\widetilde{\mathcal{V}}^m_{t_k}\}$ and $\{\widetilde{\mathcal{V}}^m_{t_k+\Delta t_k}\}$ from the self-attention layers in the last M transformer blocks at each timestep k. Here, $k \in \{n, n+1, \cdots, N\}$ and $m \in \{1, \cdots, M\}$. This process can be formulated as follows:

$$\{\widetilde{\mathcal{V}}^m_{t_k}\} = \text{Extract}\big(\boldsymbol{v}_\theta(\widetilde{\boldsymbol{Z}}_{t_k}, t_k)\big) \tag{13}$$

$$\{\widetilde{\mathcal{V}}^m_{t_k+\Delta t_k}\} = \text{Extract}\big(\boldsymbol{v}_\theta(\widetilde{\boldsymbol{Z}}_{t_k+\Delta t_k}, t_k + \Delta t_k)\big), \tag{14}$$

where $\widetilde{\boldsymbol{Z}}_{t_k}$ and $\widetilde{\boldsymbol{Z}}_{t_k+\Delta t_k}$ denote the latents during inversion for RF-Solver.

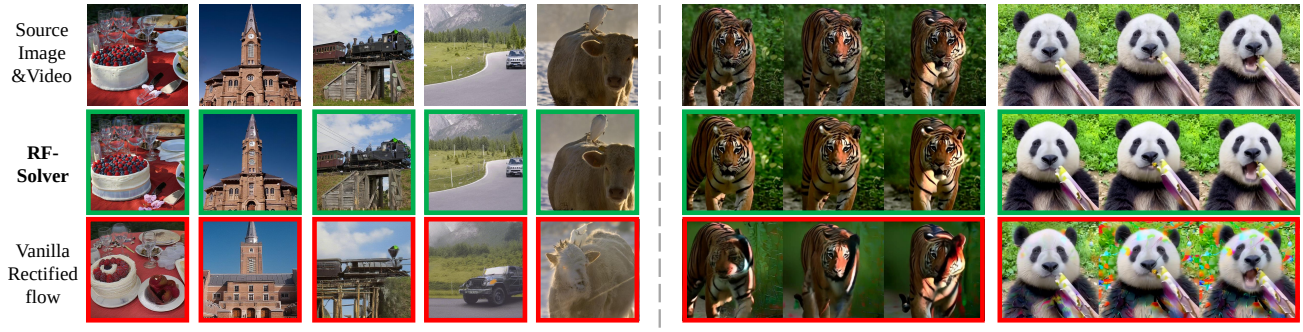
During the first n timesteps of denoising, considering the mth transformer block at the timestep k, the original self-attention in the network $\boldsymbol{v}_\theta$ can be formulated as:

$$\boldsymbol{F}^m_{t_k} = \text{Attention}(\mathcal{Q}^m_{t_k}, \mathcal{K}^m_{t_k}, \mathcal{V}^m_{t_k}), \tag{15}$$

where $\boldsymbol{F}^m_{t_k}$ denotes the output feature of the self-attention module and $\mathcal{Q}^m_{t_k}, \mathcal{K}^m_{t_k}, \mathcal{V}^m_{t_k}$ represent query, key and value for attention during the denoising process, respectively.

In RF-Edit, the above self-attention mechanism is modified to cross-attention where $\mathcal{V}^m_{t_k}$ is replaced by $\widetilde{\mathcal{V}}^m_{t_k}$,

$$\boldsymbol{F}^{m\prime}_{t_k} = \text{Attention}(\mathcal{Q}^m_{t_k}, \mathcal{K}^m_{t_k}, \widetilde{\mathcal{V}}^m_{t_k}). \tag{16}$$



(a). Image Inversion and Reconstruction (b). Video Inversion and Reconstruction

Figure 4. **Qualitative Results of Image and Video Reconstruction**. Our method (The second row) illustrates better performance compared with the rectified flow baselines (The third row) on both image and video reconstruction.

The modified output feature $\boldsymbol{F}^{m\prime}_{t_k}$ is then passed to the subsequent modules for further processing.

Similarly, this feature-sharing process is also adopted in the derivative calculation process of RF-Solver:

$$\boldsymbol{F}^{m\prime}_{t_{k+\Delta t_k}} = \text{Attention}(\mathcal{Q}^{m}_{t_{k+\Delta t_k}}, \mathcal{K}^{m}_{k+\Delta t_k}, \widetilde{\mathcal{V}}^{m}_{k+\Delta t_k}). \quad (17)$$

The proposed RF-Edit framework enables high-quality editing while preserving structural information. Building on this concept, we design two sub-modules for RF-Edit, specifically tailored for image editing and video editing. For image editing, we use FLUX [1] as the backbone, which comprises several double blocks and single blocks. Double blocks independently modulate text and image features, while single blocks concatenate these features for unified modulation. In this architecture, RF-Edit shares features within the single blocks, as they capture information from both the source image and the source prompt, enhancing the ability of the model to preserve the structural information of the source image. For video editing, we employ OpenSora [2] as the backbone. The DiT blocks in OpenSora include spatial attention, temporal attention, and text cross-attention. Within this architecture, the structural information of the source video is captured in the spatial attention module, where we implement feature sharing.

4. Experiment

4.1. Setup

Baselines. We select the vanilla Rectified Flow sampler as the primary baseline for all tasks. Additionally, for image editing, we compare our method with P2P [17], DiffEdit [12], SDEdit [33], PnP [49], Pix2pix [37] and RF-Inversion [45]. For video editing tasks, we compare our methods with FateZero [39], FLATTEN [10], COVE [51], RAVE [25], Tokenflow [15]. Detailed experimental settings of these methods are provided in the Appendix.

Implementation Details. In the experiment, we adopt the guidance-distilled variant of FLUX [1] for image tasks and

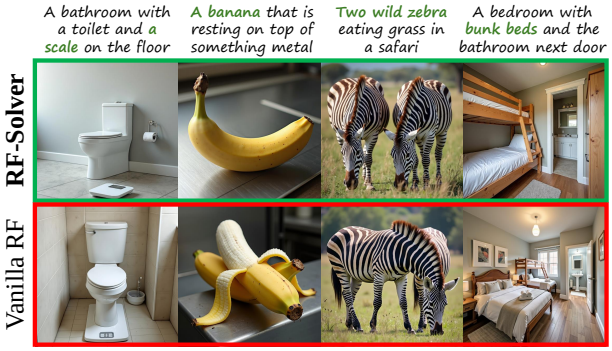


Figure 5. **Qualitative Results of Text-to-Image Generation.** With the RF-Solver, the model can generate images with higher quality (The first row) than baselines (The Second row).

Metric	FID (↓)	Clip Score (↑)
RF	26.55	31.01
Ours	**25.45**	**31.09**

Table 1. **Quantitive results on Text-to-Image Generation.** RF-Solver outperforms the vanilla RF sampler.

OpenSora [2] for video tasks. For sampling, we set the denoising step to 10 for our method. For reconstruction and editing, the inversion and denoising steps are set to 30. The derivative computation in RF-Solver requires an additional forward pass, resulting in the network needing to forward twice at each timestep. As a result, when comparing our method with the Rectified Flow baselines, *we set the number of timestep for the vanilla Rectified Flow to be* ***twice*** *that of our method* in order to maintain the same number of function evaluations (NFE) to ensure a fair comparison. Both the baseline and our method are conducted on a single A100 GPU with 40GB memory.

Evaluation Metrics For text-to-image sampling, we randomly select 10k images from the MSCOCO validation set [29] and report the FID [18] and Clip Score [40]. For the inversion and reconstruction task, we report the Mean Square Error (MSE), LPIPS [60], SSIM [54], and PSNR [24]. For image editing tasks, we report the Clip Score [40], which reflects whether the edited images align with

Figure 6. **Qualititive Comparison of Image Editing.** With RF-Solver and feature-sharing mechanism in RF-Edit, our method can successfully handle various kinds of image editing cases, outperforming the previous SOTA methods. Zoom-in for the best views.

the target prompt, and LPIPS [60], which reflects whether the edited images preserve the content of the source images. For video editing tasks, we adopt the metric proposed by [22], including Subject Consistency (SC), Motion Smoothness (MS), Aesthetic Quality (AQ), and Imaging Quality (IQ). SC and MS assess the temporal consistency of the edited video, while AQ and IQ assess the visual quality.

4.2. Text-to-image Sampling

We compare the performance of our method with the vanilla rectified flow on the text-to-image generation task. Both the quantitative (Tab. 1) and qualitative results (Fig. 5) demonstrate the superior performance of RF-Solver in fundamental T2I generation tasks, producing higher-quality images that align more closely with human cognition.

4.3. Inversion and Reconstruction

We conduct experiments on inversion and reconstruction for both image and video modalities, comparing our methods with the vanilla rectified flow sampler. For image inversion and reconstruction, we use images from the MSCOCO dataset, and for video inversion and reconstruction, we select videos from social media platforms such as TikTok and other publicly available sources [3, 4]. We use GPT-4.0 to annotate the content of the images and videos in detail, and then manually polish the GPT-generated content. These annotations are used as the source and target prompts for inversion and reconstruction.

	Mehtod	MSE (↓)	LPIPS (↓)	SSIM (↑)	PSNR (↑)
image	RF	0.0268	0.6253	0.7626	28.28
	Ours	**0.0094**	**0.4242**	**0.9271**	**29.83**
video	RF	0.0206	0.4159	0.8134	18.12
	Ours	**0.0139**	**0.3299**	**0.8805**	**18.32**

Table 2. **Quantitative Results on Inversion and Reconstruction.** Our methods significantly improve the reconstruction accuracy of both images and videos.

Quantitive Comparison. The quantitative comparisons (Tab. 2) are conducted on both images and videos to illustrate the similarity between the source and reconstruction results. Our method demonstrates superior performance across all four metrics compared with vanilla rectified flow.
Qualitative Comparison. RF-Solver effectively reduces the error in the solution of RF ODE, thereby increasing the accuracy of the reconstruction. As illustrated in Fig. 4(a), the image reconstruction results using vanilla rectified flow exhibit noticeable drift from the source image, with significant alterations to the appearance of subjects such as cake, church, train, and jeep. For video reconstruction, as shown in Fig. 4(b), the baseline reconstruction results suffer from distortion. In contrast, RF-Solver significantly alleviate these issues, achieving more satisfactory results.

4.4. Editing

We conduct experiments to evaluate the image and video editing performance of our methods. Image editing usu-

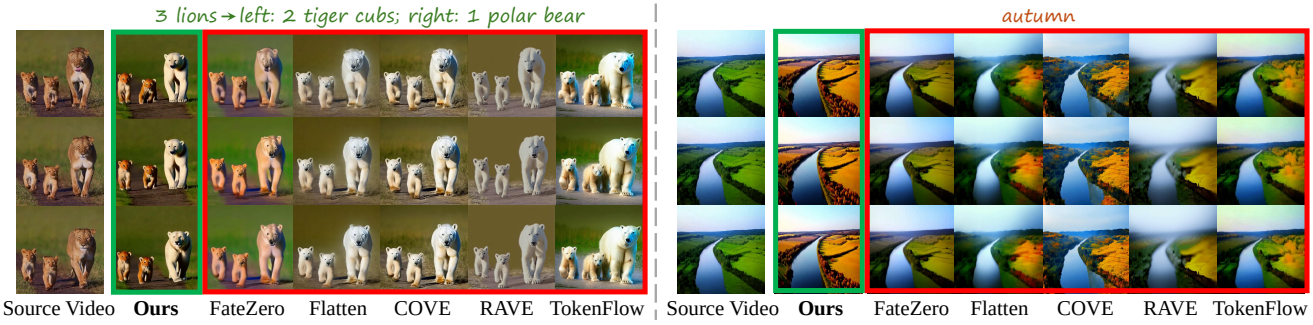


Figure 7. **Qualititive Comparison of Video Editing.** The first video has 200 frames with a resolution of 512×512, and the second video has 60 frames with a resolution of 1024×768 (we compress the frames for a neat layout). Our method outperforms various previous SOTA video editing methods, especially excelling in dealing with complex prompts.

	P2P	DiffEdit	SDEdit	PnP	Pix2Pix	RF-Inv	**Ours**
LPIPS (↓)	0.419	0.157	0.394	**0.080**	0.155	0.318	0.149
Clip Score (↑)	30.70	32.68	31.61	30.58	32.33	33.02	**33.66**

Table 3. **Quantitative Results of Image Editing.** RF-Edit can effectively edit the images according to the prompts while keeping the unrelated regions unchanged.

	FateZero	Flatten	COVE	RAVE	Tokenflow	**Ours**
SC (↑)	0.9382	0.9420	0.9433	0.9292	0.9439	**0.9501**
MS (↑)	0.9611	0.9528	0.9697	0.9519	0.9632	**0.9712**
AQ (↑)	0.6092	0.6329	0.6717	0.6586	0.6742	**0.6796**
IQ (↑)	0.6898	0.7024	0.7163	0.6917	0.7128	**0.7207**

Table 4. **Quantitative Results of Video Editing.** RF-Edit outperforms a number of previous SOTA video editing methods.

ally involves replacing the subject in the image with another one, adding new items, and global editing. For the first two types of editing, the background of the source image is expected to remain unchanged after editing. For global editing such as style transfer, the overall structure of the source image is expected to remain unchanged. Compared to image editing, video editing is much more challenging due to the complexity of modeling temporal motions in videos. Recent mainstream video editing methods usually focus on replacing the subjects and performing global editing.

Quantitative Comparison. For image editing, we perform quantitative comparison between our methods and baselines (Tab. 3), reporting the Clip score and LPIPS. Our method outperforms all other methods in Clip score, indicating that the edited images align well with the user-provided prompts. For LPIPS, it is noted that PnP [49] has a much lower value than all other methods. Based on the qualitative results (Fig. 6), it can be seen that PnP is only suitable for editing cases that do not significantly modify the structure or shape of the source image (such as changing red roses into yellow sunflowers). For shape editing, such as modifying a car into a motorbike, PnP fails to edit this case, resulting in an image very similar to the source. Consequently, although PnP has the lowest LPIPS score, its clip score is the lowest.

For video editing, we compare our methods with baseline methods using the VBench [22] metrics (Tab. 4). The results illustrate that our methods successfully maintain temporal consistency in long videos, achieving the highest Subject Consistency (SC) and Motion Smoothness (MS) scores. Additionally, our method demonstrates superior visual quality, outperforming the baselines at Aesthetic Quality (AQ) and Image Quality (IQ).

Qualitative Comparison. For image editing, we compare the performance of our methods with several baselines across different types of editing tasks (Fig. 6). The baseline methods often suffer from background changes or fail to perform the desired edits. In contrast, our methods demonstrate satisfying performance, effectively achieves a balanced trade-off between the fidelity to the target prompt and preservation of the source image. To be noticed, although RF-inversion [45] also uses the rectified flow model for image editing (third row in Fig. 6), the structure of the source image which is unrelated to editing prompt (such as background and human appearance) is modified obviously.

For video editing, we primarily evaluate the performance of our methods on long videos (200 frames) and high-resolution videos (1280×768). Furthermore, we assess the performance on complicated videos and prompts where there are multiple objects in the video, and the user has different editing requirements for each object. The qualitative results are shown in Fig. 7. Our method successfully handles complicated editing cases (*e.g.*, modifying the leftmost lion among three lions into a white polar bear and changing the other two small lions into orange tiger cubs), whereas all other baseline methods fail in this scenario. Our method also demonstrates strong performance in global editing tasks, such as transforming scenes into autumn.

4.5. Ablation Study

We conduct ablation studies to illustrate the effectiveness of RF-Solver and RF-Edit. Without loss of generality, these ablation studies are performed on the image tasks using FLUX [1] as the base model.

Taylor Expansion Order of RF-Solver. We investigated the impact of the Taylor expansion order in RF-Solver (Tab. 5) under the same NFE across different orders.

	Metric	RF	**RF-Solver-2**	RF-Solver-3
Sampling	FID (↓)	26.55	25.45	**25.37**
	Clip Score (↑)	31.01	**31.09**	31.09
Inversion	MSE (↓)	0.0268	**0.0094**	0.0131
	LPIPS (↓)	0.6253	**0.4242**	0.4817
Editing	LPIPS (↓)	0.1524	**0.1494**	0.1503
	Clip Score (↑)	32.97	**33.66**	33.18

Table 5. **Ablation Study on the Taylor Expansion Order.** We choose the 2-order expansion (*i.e.* RF-Solver-2) for the downstream tasks for its effectiveness and simplicity.

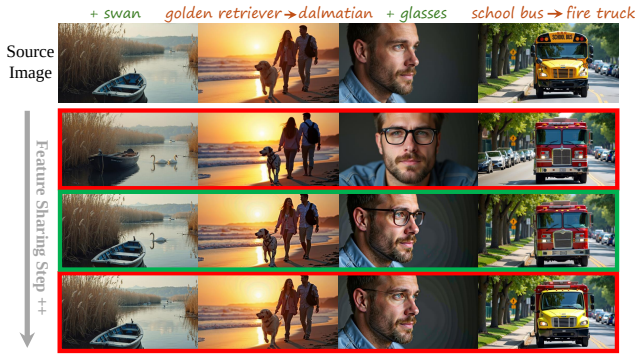


Figure 8. **Ablation Study of Feature-Sharing Step in RF-Edit.** A too-small feature-sharing step leads to inconsistency between source and target images. On the other hand, a too-large feature-sharing step can cause the failure of editing.

The second-order expansion demonstrated a significant improvement across various tasks compared to the first-order expansion (*i.e.*, the vanilla rectified flow). However, higher-order expansions did not yield further enhancements. We speculate that this is primarily due to higher-order Taylor expansions requiring more inference steps per timestep. With a fixed NFE, this results in a reduced overall number of timesteps compared to lower-order expansions, leading to suboptimal performance. Moreover, computing the higher-order derivatives of $v_\theta t_i$ substantially increases the complexity of the algorithm, posing challenges for practical applications. Consequently, we predominantly employed second-order expansion in our experiments.

Feature Sharing Steps of RF-Edit. RF-Edit leverages feature sharing to maintain the structural consistency between original images and edited images. However, an excessive number of feature-sharing steps may result in the edited output being overly similar to the source image, ultimately undermining the intended editing objectives (Fig. 8). To investigate the impact of feature-sharing steps on editing results, we incrementally increase the number of feature-sharing steps applied to the same image. Due to the varying levels of difficulty that different images presented to the model, the optimal number of sharing steps may differ across cases. Experimental results reveal that setting the sharing step to 5 effectively meets the editing requirements for most images. Additionally, we can customize the sharing step for each image to identify the most satisfying outcome.

5. Conclusion

In this paper, we propose RF-Solver, a versatile sampler for the rectified flow model that solves the rectified flow ODE with reduced error, thus enhancing the image and video generation quality across various tasks such as sampling and reconstruction. Based on RF-Solver, we further propose RF-Edit, which achieves high-quality editing performance while effectively preserving the structural information in source images or videos. Extensive experiments demonstrate the versatility and effectiveness of our methods.

References

[1] Flux. https://github.com/black-forest-labs/flux/. 2, 5, 6, 8

[2] Opensora. https://github.com/hpcaitech/Open-Sora/. 2, 5, 6

[3] Pexels. https://www.pexels.com/. Accessed: 2023-11-16. 7

[4] Pixabay. https://pixabay.com/. Accessed: 2023-11-16. 7

[5] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42 (4):1–11, 2023. 3

[6] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. Sega: Instructing text-to-image models using semantic guidance. *Advances in Neural Information Processing Systems*, 36: 25365–25389, 2023. 3

[7] 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. 3

[8] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023. 3

[9] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 3

[10] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023. 3, 6

[11] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 3

[12] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR 2023 (Eleventh International Conference on Learning Representations)*, 2023. 6

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 5

[14] Adham Elarabawy, Harish Kamath, and Samuel Denton. Direct inversion: Optimization-free text-driven real image editing with diffusion models. *arXiv preprint arXiv:2211.07825*, 2022. 3

[15] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 3, 6

[16] Jiayi Guo, Xingqian Xu, Yifan Pu, Zanlin Ni, Chaofei Wang, Manushree Vasu, Shiji Song, Gao Huang, and Humphrey Shi. Smooth diffusion: Crafting smooth latent spaces in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7548–7558, 2024. 2

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2, 3, 5, 6

[18] 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. 6

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3

[20] Nisha Huang, Fan Tang, Weiming Dong, Tong-Yee Lee, and Changsheng Xu. Region-aware diffusion for zero-shot text-driven image editing. *arXiv preprint arXiv:2302.11797*, 2023. 3

[21] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525*, 2024. 3

[22] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 7, 8

[23] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024. 3

[24] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6

[25] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6507–6516, 2024. 3, 6

[26] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 3

[27] Yao-Chih Lee, Ji-Ze Genevieve Jang, Yi-Ting Chen, Elizabeth Qiu, and Jia-Bin Huang. Shape-aware text-driven layered video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14317–14326, 2023. 3

[28] Shanglin Li, Bohan Zeng, Yutang Feng, Sicheng Gao, Xiuhui Liu, Jiaming Liu, Lin Li, Xu Tang, Yao Hu, Jianzhuang Liu, et al. Zone: Zero-shot instruction-guided local editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6254–6263, 2024. 3

[29] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[30] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8599–8608, 2024. 3

[31] 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. 2

[32] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022. 3

[33] 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. In *International Conference on Learning Representations*, 2022. 6

[34] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 3

[35] 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. 2, 3

[36] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8089–8099, 2024. 3

[37] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3, 6

[38] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2

[39] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15932–15942, 2023. 3, 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[41] Hareesh Ravi, Sachin Kelkar, Midhun Harikumar, and Ajinkya Kale. Preditor: Text guided image editing with diffusion prior. *arXiv preprint arXiv:2302.07979*, 2023. 3

[42] 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*, pages 10684–10695, 2022. 2

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[44] Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[45] L Rout, Y Chen, N Ruiz, C Caramanis, S Shakkottai, and W Chu. Semantic image inversion and editing using rectified stochastic differential equations. 2024. 2, 3, 5, 6, 8

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3, 5

[47] 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. 3

[48] Wenhao Sun, Rong-Cheng Tu, Jingyi Liao, and Dacheng Tao. Diffusion model-based video editing: A survey. *arXiv preprint arXiv:2407.07111*, 2024. 3

[49] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 3, 5, 6, 8

[50] 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. 3

[51] Jiangshan Wang, Yue Ma, Jiayi Guo, Yicheng Xiao, Gao Huang, and Xiu Li. Cove: Unleashing the diffusion feature correspondence for consistent video editing. *arXiv preprint arXiv:2406.08850*, 2024. 3, 6

[52] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*, 2023. 3

[53] Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023. 3

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[55] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 3

[56] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Fresco: Spatial-temporal correspondence for zero-shot video translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8703–8712, 2024. 3

[57] Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *arXiv preprint arXiv:2406.03293*, 2024. 3

[58] Zhen Yang, Ganggui Ding, Wen Wang, Hao Chen, Bohan Zhuang, and Chunhua Shen. Object-aware inversion and reassembly for image editing. *arXiv preprint arXiv:2310.12149*, 2023. 3

[59] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[60] 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*, pages 586–595, 2018. 6, 7

[61] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023. 3