Text-to-Image Rectified Flow as Plug-and-Play Priors

Xiaofeng Yang [1] Cheng Chen [1] Xulei Yang [2] Fayao Liu [2] Guosheng Lin [1]

[1]College of Computing and Data Science, Nanyang Technological University, Singapore [2]Institute for Infocomm Research, A*STAR, Singapore

SCAN ME

Introduction

Large-scale diffusion models have achieved remarkable performance in generative tasks. Beyond their initial training applications, these models have proven their ability to function as versatile plug-and-play priors. For instance, 2D diffusion models can serve as loss functions to optimize 3D implicit models. Rectified Flow, a novel class of generative models, has demonstrated superior performance across various domains. Compared to diffusion-based methods, rectified flow approaches surpass them in terms of generation quality and efficiency. In this work, we present theoretical and experimental evidence demonstrating that rectified flow based methods offer similar functionalities to diffusion models — they can also serve as effective priors. Besides the generative capabilities of diffusion priors, motivated by the unique time-symmetry properties of rectified flow models, a variant of our method can additionally perform image inversion. Experimentally, our rectified flow based priors outperform their diffusion counterparts — the SDS and VSD losses — in text-to-3D generation. Our method also displays competitive performance in image inversion and editing.

TL;DR: Distill knowledge from 2D flow-matching models for 3D generation and 2D editing.

Our Methods

1), Baseline: RFDS – equavalent to Score Distillation Sampling

We are interested in finding the gradient with respect to the input given a trained flow matching model. Starting from the training loss of flow matching:

$$\int_0^1 \mathbb{E}\left[\| \boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon} \|^2\right] \mathrm{d}t, \text{ with } \boldsymbol{x} = g(\boldsymbol{\theta}) \text{ and } \boldsymbol{x_t} = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}. \tag{1}$$

To find $\boldsymbol{\theta}$, the gradients of the above equation with respect to $\boldsymbol{\theta}$ can be written as:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{rfds}}(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}, t) = 2 \times \mathbb{E}\left[\underbrace{(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}_{\text{Flow Residual}} \frac{\partial(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}{\partial \boldsymbol{x}} \underbrace{\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}}_{\text{Generator Jacobian}}\right]. \tag{2}$$

It can be further simplified as:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{rfds}}(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}, t) = 2 \times \mathbb{E}\left[\underbrace{(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}_{\text{Flow Residual}} \underbrace{\frac{\partial(-\dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}{\partial \boldsymbol{x}}}_{-\dot{\alpha}_t} + \underbrace{\frac{\partial \boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t)}{\partial \boldsymbol{x_t}}}_{\text{Network Jacobian}} \underbrace{\frac{\partial \boldsymbol{x_t}}{\partial \boldsymbol{x}}}_{\alpha_t} \underbrace{\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}}_{\text{Generator Jacobian}}\right]. \tag{3}$$

Finally:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{rfds}}(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}, t) \simeq \mathbb{E}\left[w(t) \underbrace{(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}_{\text{Flow Residual}} \underbrace{\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}}_{\text{Generator Jacobian}}\right]. \tag{4}$$

2), inverse RFDS (iRFDS) – find the noise instead of the image – for image editing:

$$\nabla_{\boldsymbol{\epsilon}} \mathcal{L}_{\text{irfds}}(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}, t) = 2 \times \mathbb{E}\left[\underbrace{(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}_{\text{Flow Residual}} \underbrace{\frac{\partial(-\dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}}}_{-\dot{\sigma}_t} + \underbrace{\frac{\partial \boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t)}{\partial \boldsymbol{x_t}}}_{\text{Network Jacobian}} \underbrace{\frac{\partial \boldsymbol{x_t}}{\partial \boldsymbol{\epsilon}}}_{\sigma_t}\right]. \tag{5}$$

Then

$$\nabla_{\boldsymbol{\epsilon}} \mathcal{L}_{\text{irfds}}(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}, t) \simeq \mathbb{E}\left[w'(t) \underbrace{(\boldsymbol{v}_{\boldsymbol{\phi}}(\boldsymbol{x_t}, t) - \dot{\alpha}_t \boldsymbol{x} - \dot{\sigma}_t \boldsymbol{\epsilon})}_{\text{Flow Residual}}\right]. \tag{6}$$

3), RFDS Reversal (RFDS-Rev) – improved RFDS

We propose a novel method that improves upon the baseline RFDS by iteratively applying RFDS for image optimization and iRFDS for image inversion. Please refer to our paper for detailed derivations.

Algorithm 1: The RFDS-Rev Algorithm.

Initialize the learnable parameter $\boldsymbol{\theta}$
while *Not Converge* **do**
  Sample random timestep t
  Sample random noise ϵ
  for *n steps* **do**
    Freeze $\boldsymbol{\theta}$, optimize ϵ with iRFDS (Eq. 6)
  Optimize $\boldsymbol{\theta}$ with optimized ϵ based on RFDS (Eq. 4)
RETURN θ

Our Results

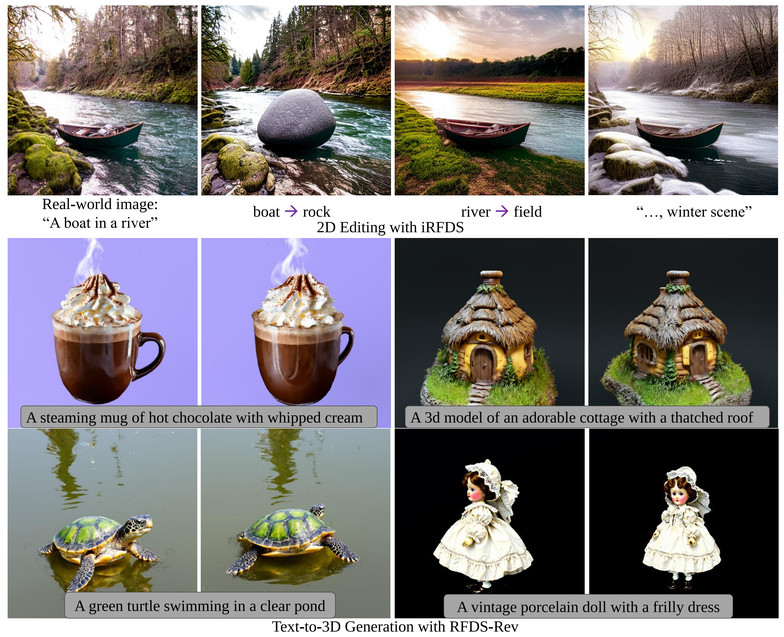

Figure 1. Demonstration of potential use cases of the proposed rectified flow priors. Our methods can be used on 2D inversion and editing and text-to-3D generation.

Additional Results

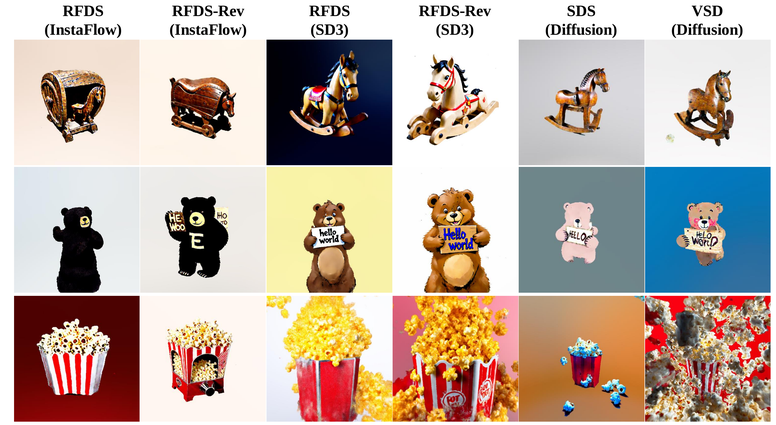

Figure 2. 3D Comparison. Text Prompts: "An antique wooden rocking horse", "A bear holding a sign that says Hello world" and "Hot popcorn jump out from the red striped popcorn maker".

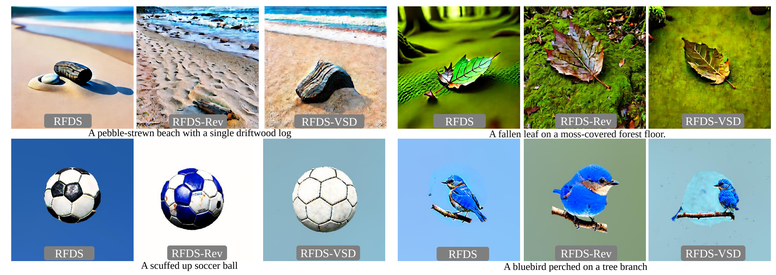

Figure 3. Ablation experiments of RFDS-Rev vs. RFDS-VSD. Top: 2D case. Bottom: 3D case.

Table 1. Results of Text-to-3D on T^{3}Bench dataset. Our proposed methods achieve state-of-the-art performance among 2D lifting methods, beating the diffusion based SDS and VSD priors.

Method	Dreamfusion (SDS)	ProlificDreamer (VSD)	**RFDS**	**RFDS-Rev**	**RFDS**	**RFDS-Rev**
Base Model	Diffusion	Diffusion	InstaFlow	InstaFlow	SD3	SD3
Single Object	24.9	51.1	46.9	57.6	49.4	**59.8**
Surroundings	19.3	42.5	34.5	44.6	39.3	**54.5**
Multiple Objects	17.3	45.7	28.3	44.6	42.9	**55.2**
Average	20.5	46.4	36.5	48.9	43.9	**56.5**