# Smooth Diffusion: Crafting Smooth Latent Spaces in Diffusion Models

Supplementary Material

## A. Implementation Details

This section elaborates on details briefly introduced in the main paper. These include the notation, the basic training objective, the interpolation standard deviation (ISTD) metric, and our utilization of Null-text inversion (NTI) [5] for real-image interpolation.

### A.1. Notation

Stable Diffusion [6] employs an efficient "latent" diffusion pipeline. Here the "latent" refers to using an individually trained (VAE) [3] to compress an input image $\boldsymbol{x}_0$ into its VAE-space representation $\boldsymbol{z}_0$:

$$\boldsymbol{z}_0 = \mathcal{E}(\boldsymbol{x}_0), \quad \boldsymbol{x}_0 = \mathcal{D}(\boldsymbol{z}_0), \tag{1}$$

where $\mathcal{E}$ and $\mathcal{D}$ represent the encoder and decoder of the VAE, respectively. For simplicity, we exclude this conversion process and only use "$\boldsymbol{x}$"-based notations in the main paper. Although we chose Stable Diffusion as our baseline due to its popularity and high performance, our training pipeline is not specifically tailored for latent diffusion models and is compatible with other diffusion models.

### A.2. Basic Training Objective

Smooth Diffusion's training objective comprises two key components: 1) a basic training objective primarily centered on noise prediction but flexible in formulation for different diffusion models, and 2) our proposed Step-wise Variation Regularization term. In our experiments, the basic training objective is:

$$\mathcal{L}_{\text{base}} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon},t}\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{x}_t, t)\|_2^2, \tag{2}$$

which is a commonly adopted training objective across many diffusion models, *e.g.*, Stable Diffusion [6].

### A.3. ISTD

The goal of ISTD is to quantify the deviation of pixel-space changes given the same fixed-step changes in latent space. A lower deviation implies the input latents and output images are more likely to change smoothly. In our experiments, we first randomly draw 500 text prompts from the MS-COCO validation set [4]. For each prompt, we then sample two random Gaussian noises, $\boldsymbol{\epsilon}^a$ and $\boldsymbol{\epsilon}^b$. Next, we execute uniform spherical linear interpolations (slerp) between $\boldsymbol{\epsilon}^a$ and $\boldsymbol{\epsilon}^b$ for 11 times, varying the mixing ratio $\eta$ from 0 to 1:

$$\boldsymbol{\epsilon}^\eta = \text{slerp}(\boldsymbol{\epsilon}^a, \boldsymbol{\epsilon}^b, \eta), \quad \eta = 0, 0.1, 0.2, \cdots, 1. \tag{3}$$

We employ the testing diffusion model to generate 11 interpolated images $\{\widehat{\boldsymbol{x}_0^\eta}\}_{\eta=0}^1$ from $\{\boldsymbol{\epsilon}^\eta\}_{\eta=0}^1$. Notice that Eq. 3 guarantees that the latent space changes between every two adjacent latents (*i.e.*, $\boldsymbol{\epsilon}^\eta$ and $\boldsymbol{\epsilon}^{\eta+0.1}$) are the same. Hence, we calculate the L2 distances between every two adjacent images (*i.e.*, $\widehat{\boldsymbol{x}_0^\eta}$ and $\widehat{\boldsymbol{x}_0^{\eta+0.1}}$) and compute the standard deviation of these distances. Finally, ISTD is the average of standard deviations over 500 different text prompts. For a fair comparison, the text prompts and the noises for each prompt are the same for different testing models.

### A.4. NTI for real-image interpolation

NTI is initially designed to transform a real image $\boldsymbol{x}_0$ into a latent $\widetilde{\boldsymbol{x}_T}$, along with a series of learnable null-text embeddings $\{\varnothing_t\}_{t=1}^T$ for each step $t$. The optimization for each $\varnothing_t$ is formulated as:

$$\min_{\varnothing_t} \|\widetilde{\boldsymbol{x}_{t-1}} - \text{DDIM}(\widetilde{\boldsymbol{x}_t}, t, \xi, \varnothing_t)\|_2^2. \tag{4}$$

where $\{\widetilde{\boldsymbol{x}_t}\}_{t=1}^T$ represents intermidiate noisy images estimated by DDIM inversion [8]. For simplicity, $\text{DDIM}(\widetilde{\boldsymbol{x}_t}, t, \xi, \varnothing_t)$ denotes the DDIM sampling process at step $t$, utilizing the text embedding $\xi$, the null-text embedding $\varnothing_t$ and the classifier-free guidance scale $w = 7.5$.

For real-image interpolation, we optimize a shared series of $\{\varnothing_t\}_{t=1}^T$ for two real images, $\boldsymbol{x}_0^a$ and $\boldsymbol{x}_0^b$:

$$\begin{aligned}\min_{\varnothing_t} &\|\widetilde{\boldsymbol{x}_{t-1}^a} - \text{DDIM}(\widetilde{\boldsymbol{x}_t^a}, t, \xi, \varnothing_t)\|_2^2 + \\ &\|\widetilde{\boldsymbol{x}_{t-1}^b} - \text{DDIM}(\widetilde{\boldsymbol{x}_t^b}, t, \xi, \varnothing_t)\|_2^2.\end{aligned} \tag{5}$$

In our experiments, we only interpolate the latents $\widetilde{\boldsymbol{x}_T^a}$ and $\widetilde{\boldsymbol{x}_T^b}$ following Eq. 3 and use the same null-text embeddings $\{\varnothing_t\}_{t=1}^T$ for all interpolated images.

## B. Additional Results

This section provides additional visual results of Smooth Diffusion. We display image interpolation results in Fig. 1 and Fig. 2, image inversion and reconstruction results in Fig. 3, and image editing results in Fig. 4.

**Reusability.** The LoRA component of Smooth Diffusion remains adaptable to other models sharing the same architecture as Stable Diffusion. However, the effectiveness of this reusability is not guaranteed. We evaluate the integration of this LoRA component into two popular community models, RealisticVision-V2 [2] and OpenJourney-V4 [1]. As depicted in Fig. 2, this integration also enhances the latent space smoothness of these models. This reusability makes our method eliminate the need for repeated training and become a plug-and-play module across various models.

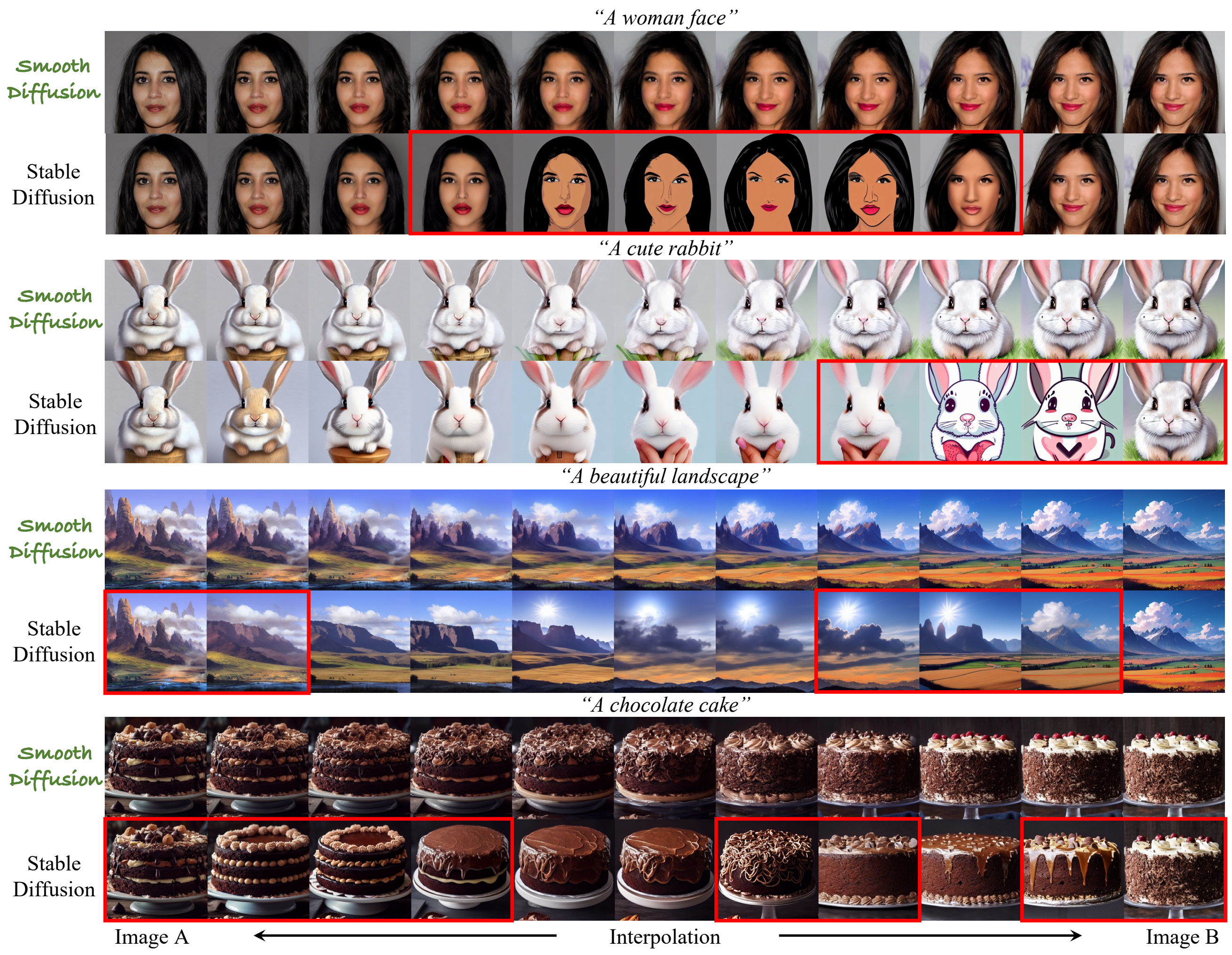

Figure 1. **Additional image interpolation results with Smooth Diffusion.** For Smooth Diffusion and Stable Diffusion [6], real images (Image A and B) are inverted into latents using Null-text inversion [5]. We perform spherical linear interpolations between latents and concatenate the resulting images as a transition sequence.

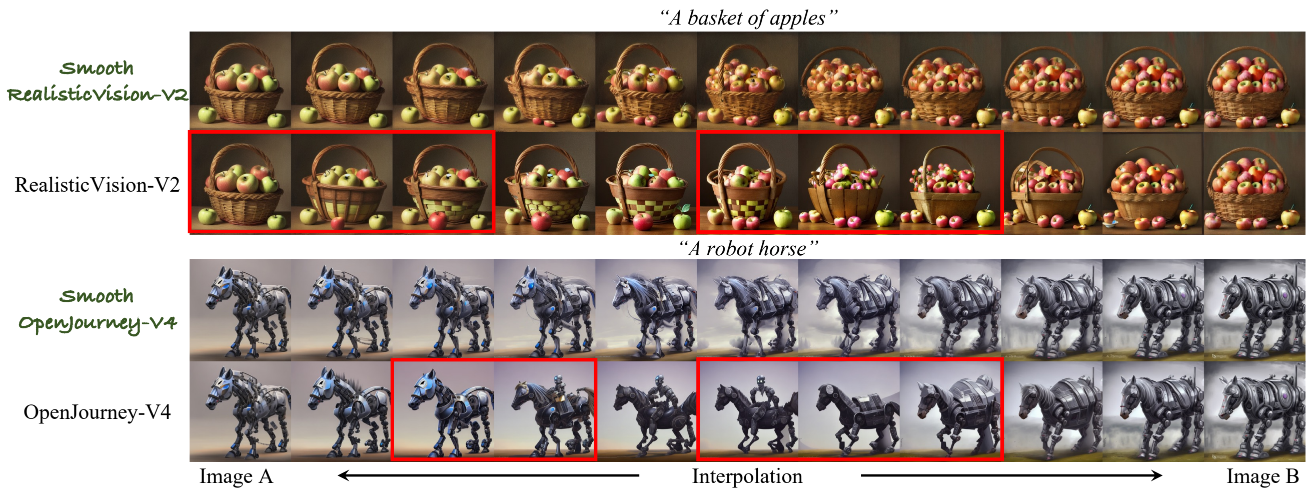

Figure 2. **Image interpolation results with community models.** We apply the LoRA component of Smooth Diffusion to RealisticVision-V2 [2] and OpenJournery-V4 [1] and perform spherical linear interpolations in their latent spaces.

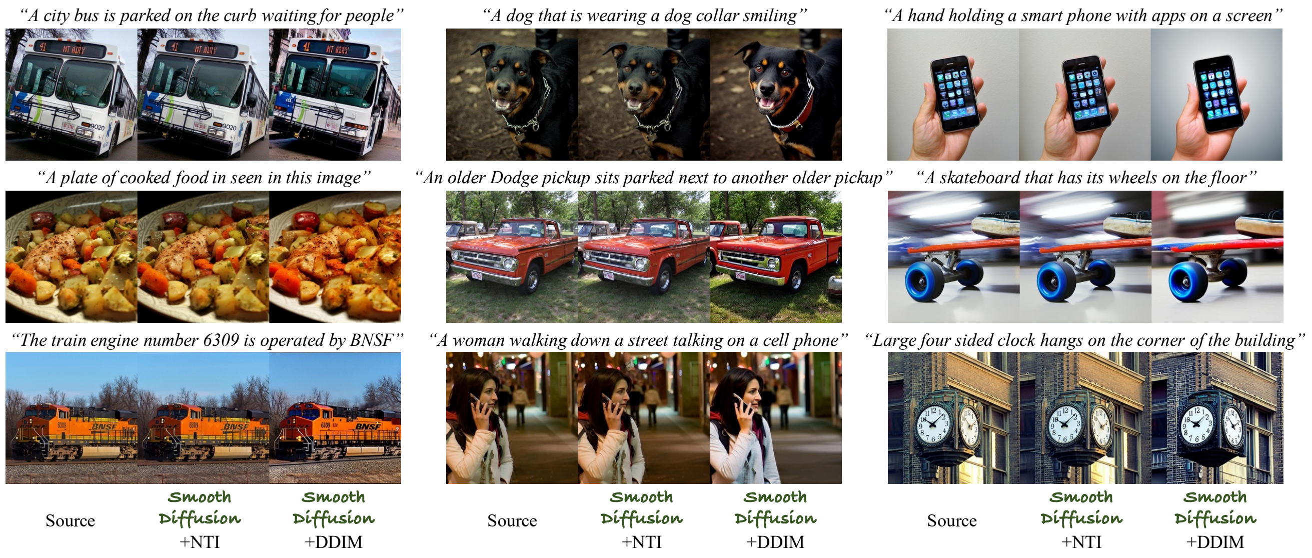
"A city bus is parked on the curb waiting for people"
"A dog that is wearing a dog collar smiling"
"A hand holding a smart phone with apps on a screen"
"A plate of cooked food in seen in this image"
"An older Dodge pickup sits parked next to another older pickup"
"A skateboard that has its wheels on the floor"
"The train engine number 6309 is operated by BNSF"
"A woman walking down a street talking on a cell phone"
"Large four sided clock hangs on the corner of the building"
Source
Smooth Diffusion +NTI
Smooth Diffusion +DDIM

Figure 3. **Additional image inversion and reconstruction results with Smooth Diffusion.** We integrate Smooth Diffusion with two typical diffusion inversion techniques, Null-text inversion [5] and DDIM inversion [8].

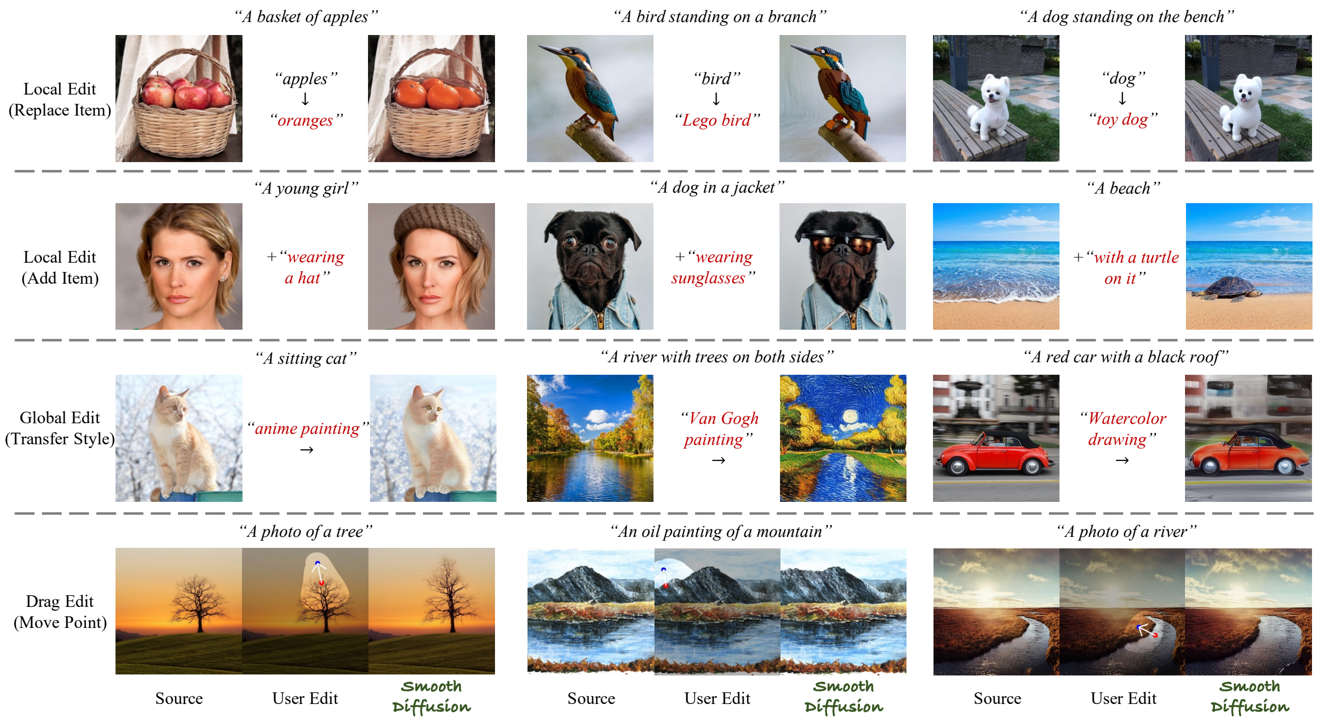

Figure 4. **Additional image editing results with Smooth Diffusion.** Both text-based image editing and drag-based image editing are evaluated. For text-based image editing, we consider both local and global edits to test Smooth Diffusion. For drag-based image editing, Smooth Diffusion is integrated into the framework of DragDiffusion [7].

## References

[1] OpenJourney-V4. https://huggingface.co/prompthero/openjourney-v4, 2023. 1, 3

[2] RealisticVision-V2. https://huggingface.co/SG161222/Realistic_Vision_V2.0, 2023. 1, 3

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2015. 1

[4] 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 *ECCV*, 2014. 1

[5] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023. 1, 2, 3

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2

[7] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. DragDiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv:2306.14435*, 2023. 4

[8] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 1, 3