FluxSpace: Disentangled Semantic Editing in Rectified Flow Models

Supplementary Material

A. Joint Transformer Block Architecture

In our editing algorithm introduced in Sec. 4, we focus on joint transformer blocks in Flux, where text and image inputs are processed in their respective representation spaces. As noted in [15], this design enables each modality to operate within its own feature space, while an interaction mechanism - specifically, the attention layer in the MM-DiT architecture - combines the two. Unlike single transformer blocks, which do not distinguish between feature spaces of different modalities, joint transformer blocks explicitly assume distinct spaces for text and image representations. Based on this difference, we conceptualize these blocks as the areas in which text content semantically influences image content. Consequently, we define our edits within these blocks, leveraging their unique capacity for cross-modal integration. Below, we outline the key components of the joint transformer block architecture in Flux to provide a better overview of our editing approach.

Modulation Mechanism. The MM-DiT architecture, based on the DiT framework [33], begins by performing a modulation operation utilizing coarse conditioning c_{pool}, timestep embedding t_{emb} and guidance scale embedding g_{emb}. These embeddings are combined to compute the modulation embedding m, as described in Eq. 11, within the MM-DiT architecture used in Flux.

$$m = c_{pool} + t_{emb} + g_{emb} \tag{11}$$

Attention Computation. Given the modulation embedding m, input image features x, and contextual text features c_{ctxt}, the features undergo a modulation operation before the attention process. Subsequently, the text and image features are normalized using their respective layers. Using these normalized features, the joint attention block computes the attention for both modalities, using the query (Q), key (K), and value (V) features for the text and image inputs. Throughout this paper, we refer to the combination of modulation, normalization, and attention computation as a single pass through the attention layer, parameterized by θ, denoted as $l_\theta(x, c, t)$ for timestep t.

B. User Study Details

To perceptually evaluate our method against competing approaches, we conducted a user study with 50 participants on the Prolific platform, where the results are provided in Table 1. For further clarification of the user study conducted, we provide a sample question in Fig. 7. In each question, we ask the users to rate the edit on a scale of 1-to-5 (1 for unsatisfactory, 5 for very satisfactory), after providing the image before editing and the edited image.

C. Details on Quantitative Comparisons

In this section, we provide the details of the quantitative results provided in Table 1. For clarity, we explain the setup used for each of the compared approaches, the models used for evaluation, and the hyperparameters used for editing with *FluxSpace*.

C.1. Competing Methods

We compare the editing capabilities of *FluxSpace* with RF-Inversion [38], Sliders-FLUX [17], TurboEdit [12] and LEDITS++ [7]. Below, we explain the setup used for each of these methods along with implementation details where necessary.

RF-Inversion [38]. Using the algorithm provided in [38], we re-implement RF-Inversion using `diffusers` library. For the editing hyperparameters, we use the parameter set provided by the authors for the eyeglasses edit. Even though the authors do not provide the exact hyperparameters for the smile edit in their paper, we utilize the hyperparameters used for the age editing task. Specifically, for the hyperparameters starting time (s), stopping time (τ), and strength (η) we use the values **6, 25, 0.7** for the eyeglasses edit and **0, 5, 1.0** for the smile edit. For all generations, we use 30 steps, consistent with the setup used for *FluxSpace*. Comparing the results presented in Table 1 and the results provided in [38], our reported results are consistent with their quantitative evaluation, where we consider our implementation successful. As we also demonstrate qualitatively in Fig. 4 and 5, our approach succeeds over this baseline by improving the disentanglement and editing capabilities, where both approaches use `FLUX.1-dev` as the generative model.

Sliders-FLUX [17]. Even though Concept Sliders was originally developed for UNet-based diffusion models [34, 37], we utilize the extension of this method on rectified flow transformers. To do so, we use the `sliders` implementation provided for Flux, by the authors[2]. In all of our experiments, we use *text* sliders which we train for eyeglasses and smile attributes. For training, we follow the default hyperparameters provided in the official implementation, where

[2] https://github.com/rohitgandikota/sliders/tree/main/flux-sliders

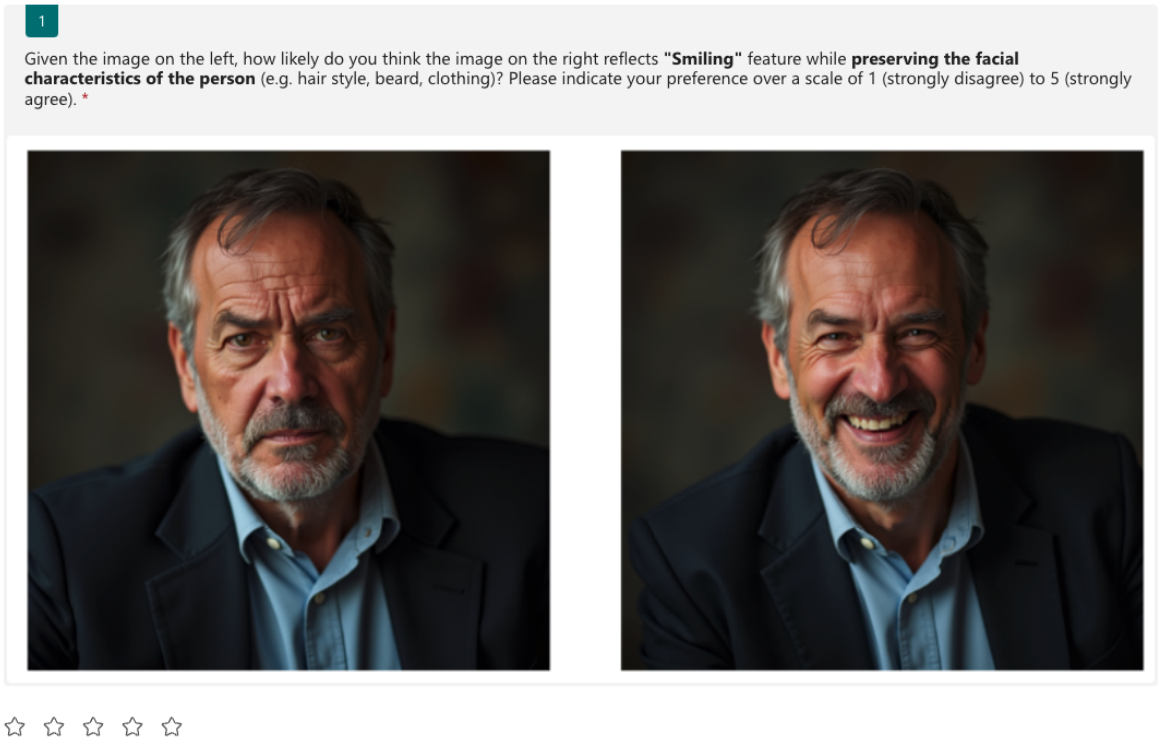

Figure 7. **User Study Setup.** We conduct our user study on unedited-edited image pairs. For each editing method, we provide the original image where the edit is not applied, with the edited image, and ask the users to rate the edit from a scale of 1-to-5. On the Likert scale that the users are asked to provide their preference on, 1 corresponds to unsatisfactory editing and 5 corresponds to a satisfactory edit.

the LoRA rank is set to 16 and training is performed for 1000 iterations for all experiments, using `FLUX.1-dev`. Demonstrated in Fig. 4 and Table 1, Sliders-FLUX struggles with the preservation of the input subject identity where significant alterations are observed during editing (see Fig. 4, middle row). We relate this issue with the training performed for the LoRA adapters and the fact that the edited concept cannot be clearly isolated. Note that our method does not require such training for the image editing task.

TurboEdit [12]. As a method based on few-step text-to-image generation models, we perform comparisons with TurboEdit [12] which uses SDXL-Turbo [39]. Following the official implementation of TurboEdit, we perform our comparisons with resolution 512 x 512 and 4 inference steps. Regarding the hyperparameters of the method, we use the pseudo-guidance scale w as 1.5, and the random seed for edits as 2[3]. Demonstrated in results presented in Fig. 4, TurboEdit succeeds in performing the edit but fails in disentanglement, where the edits "age" and "eyeglasses" result in significant edit-irrelevant changes (e.g. skin color in age edit).

LEDITS++ [7]. We also compare our method with LEDITS++ [7], which is the state-of-the-art semantic editing method for text-to-image diffusion models. In our comparisons, we use the version of LEDITS++ that uses SDXL[4]. For all of our experiments, we use the editing guidance scale and the editing threshold values **5.0** and **0.75**, as we found the guidance scale effective for eyeglasses and smile edit and we do not change the threshold value from the default setup. Despite performing well in identity preservation and semantic editing tasks, LEDITS++ results in artifacts in the edited images, which are not clearly identified in the quantitative evaluation, but acknowledged in the user study conducted.

C.2. Hyperparameter Selection

We present the quantitative results for our method in Table 1. In our experiments, we use a fixed set of hyperparameters for each edit evaluated, which are coarse editing scale λ_{coarse}, fine-grained editing scale λ_{fine}, mask threshold

[3] https://github.com/GiilDe/turbo-edit/blob/master/main.py

[4] https://github.com/ml-research/ledits_pp/tree/main

τ_m, and starting iteration for edit i. The hyperparameter sets we used for the "eyeglasses" and "smile" edits are as follows:

- **Eyeglasses:** $\lambda_{coarse} = 0.8$, $\lambda_{fine} = 5$, $\tau_m = 0.5$, $i = 3$
- **Smile:** $\lambda_{coarse} = 0.5$, $\lambda_{fine} = 8$, $\tau_m = 0.5$, $i = 5$

We use the editing prompts "eyeglasses" and "smiling" to perform these edits. Note that even though these edits require fine-grained changes in the image, our approach can apply these edits in a disentangled way even when the editing process starts in early timesteps. For all of our experiments, we set the number of inference steps as 30. We use a fixed seed of 0 in all of our experiments.

D. Supplementary Quantitative Results

In addition to the quantitative comparisons on generated images presented in Table 1, we present supplementary quantitative results on real image editing, where we compare with RF-Inversion [38]. Following their benchmark, we report "smile" editing results on the SFHQ dataset [4]. In our evaluations, we compare the two methods on DINO, CLIP-I (image-to-image similarity), CLIP-T (text-to-image similarity), Face Rec., and Runtime, following the protocol introduced in [38]. Since our method integrates the inversion algorithm proposed by RF-Inversion [38] and replaces prompt-based editing with our method, we report the benchmark metrics on both the inverted images and the input images, which we show as (I) and (R) in Table 2. Note that for DINO, CLIP-I and CLIP-T higher scores lead to better performance, whereas lower Face Rec. and runtime measurements lead to better performance. Our approach consistently performs better in terms of preserving the input details, while performing comparably in text-to-image similarity. Furthermore, we also demonstrate the flexibility of our method in integration with inversion methods.

D.1. Metrics Used

As we also specify in Table 1, we use CLIP-T, CLIP-I, and DINO metrics to quantitatively evaluate our method. To enable the reproducibility of our experiments, we also share details on the model weights we use to obtain these scores. Specifically, we use `CLIP ViT-bigG/14`[5] variant of the CLIP model to calculate the CLIP-I and CLIP-T scores. For DINO scores, we use `DINOv2`[6].

E. Supplementary Qualitative Results

In this section, we provide supplementary qualitative results to further demonstrate the capabilities of our method.

[5] https://huggingface.co/laion/CLIP-ViT-bigG-14-laion2B-39B-b160k

[6] https://huggingface.co/facebook/dinov2-base

Supplementary Comparisons. In addition to the comparisons provided in the main paper, we also compare the editing capabilities of *FluxSpace* with methods based on Stable Diffusion. We compare our method with Prompt2Prompt [19] and PnP-Diffusion [40] as competing approaches, where we perform inversion with Null-Text inversion [28] for Prompt2Prompt. We provide qualitative comparisons in Fig. 8.

Editing Examples. Supplementary to the results provided in the main paper, we provide additional editing results in the supplementary material. We provide additional results for "gender" and "sunglasses" edits in Figs. 9 and 10 for portrait images and images in natural settings. Furthermore, we provide editing results for various concepts in Fig. 11 and with multiple subjects in Fig. 12.

Attention Mask Visualizations. To demonstrate the semantic localization capability of the masks we extract to constrain the edits within the regions of interest, we provide visualizations of the attention masks in Fig. 13. For each example, we provide the original editing and its corresponding edit, coupled with the attention mask $M'_{0,edit}$, which is the mask before the thresholding operation. As we demonstrate in the provided examples, our approach succeeds in localizing the regions related to the corresponding edit in generation time. In addition to the input-edit pair and the corresponding mask, we also provide the generation and edit condition (c, c_e) under each example.

Supplementary Ablations. In addition to the ablations provided in human subjects in Fig. 6, we provide additional results as part of the supplementary material. Specifically, we provide ablations on the masking threshold τ_m in Fig. 14, editing timesteps in Fig. 15 and coarse/fine editing scales in Fig. 16. We provide the generation and edit conditions (c, c_e) under each example.

Method	DINO(R)	DINO(I)	CLIP-I(R)	CLIP-I(I)	Face Rec. (R)	Face Rec. (I)	CLIP-T	Time(s)
RF-Inversion [38]	0.710	0.917	0.683	0.921	**0.456**	0.077	**0.333**	**33**
Ours	**0.724**	**0.937**	**0.691**	**0.936**	0.475	**0.063**	0.322	41

Table 2. **Comparisons with RF-Inversion [38] on SFHQ dataset.** In addition to comparisons over generated images (see Table 1), we provide additional quantitative comparisons with RF-Inversion[38], where we integrate their inversion pipeline and replace their editing method with ours. We report quantitative results for "smile" editing task on SFHQ dataset. Note that lower is better for runtime and Face Rec. scores, and higher is better for the remaining metrics.

Figure 8. **Additional Qualitative Comparisons.** In addition to comparisons provided in the main paper, we provide additional comparisons with Prompt2Prompt [19] (with Null-Text Inversion [28]) and PnP-Diffusion [40], as Stable Diffusion based editing methods. As we demonstrate qualitatively, FluxSpace both achieves disentangled and semantically correct edits where competing methods contain artifacts in edited results (see the edit "Eyeglasses" for both methods), and significantly alter the subject identity (see "Age" edit).

Figure 9. **Gender Editing Results.** We provide additional editing results for editing the gender semantics. As shown in the examples, our method succeeds in both male-to-female and female-to-male translations. We provide editing results on both portrait images, where our edits preserve the facial details, and edits on complex scenes where we succeed in only editing the human subject. Both in terms of preserving the identity of the subject and the background details, *FluxSpace* succeeds in the disentanglement editing task.

Figure 10. **Sunglasses Editing Results.** We provide additional qualitative results for the edit "adding sunglasses". As we demonstrate on human subjects in both portrait images and more complex scenes, our editing method can accurately target where the edit should be applied without any input mask. We show the editing capabilities of *FluxSpace* both in images where the human subject is the main focus of the image (first two rows) and with human subjects as a part of a scene (last two rows). In both cases, our method succeeds in performing the desired edit and preserving the edit-irrelevant details.

Figure 11. **Conceptual Editing Results.** We provide editing results with abstract concepts, that affect the overall appearance of the image. Our method succeeds in performing edits that alter the content of the image (top row) by being able to interpret the structures in the unedited image (e.g. the trees on the back for the edit "cherry blossom") and can change the style and overall appearance of the image (bottom row).

Figure 12. **Editing Results with Multiple Subjects.** We present qualitative results on images with multiple subjects. In addition to images with only one subject to be edited, *FluxSpace* can apply edits by identifying semantics globally and editing multiple subjects at the same time. Note that our method does not use any external mask, and performs the edit completely with the semantic understanding of the rectified flow transformer.

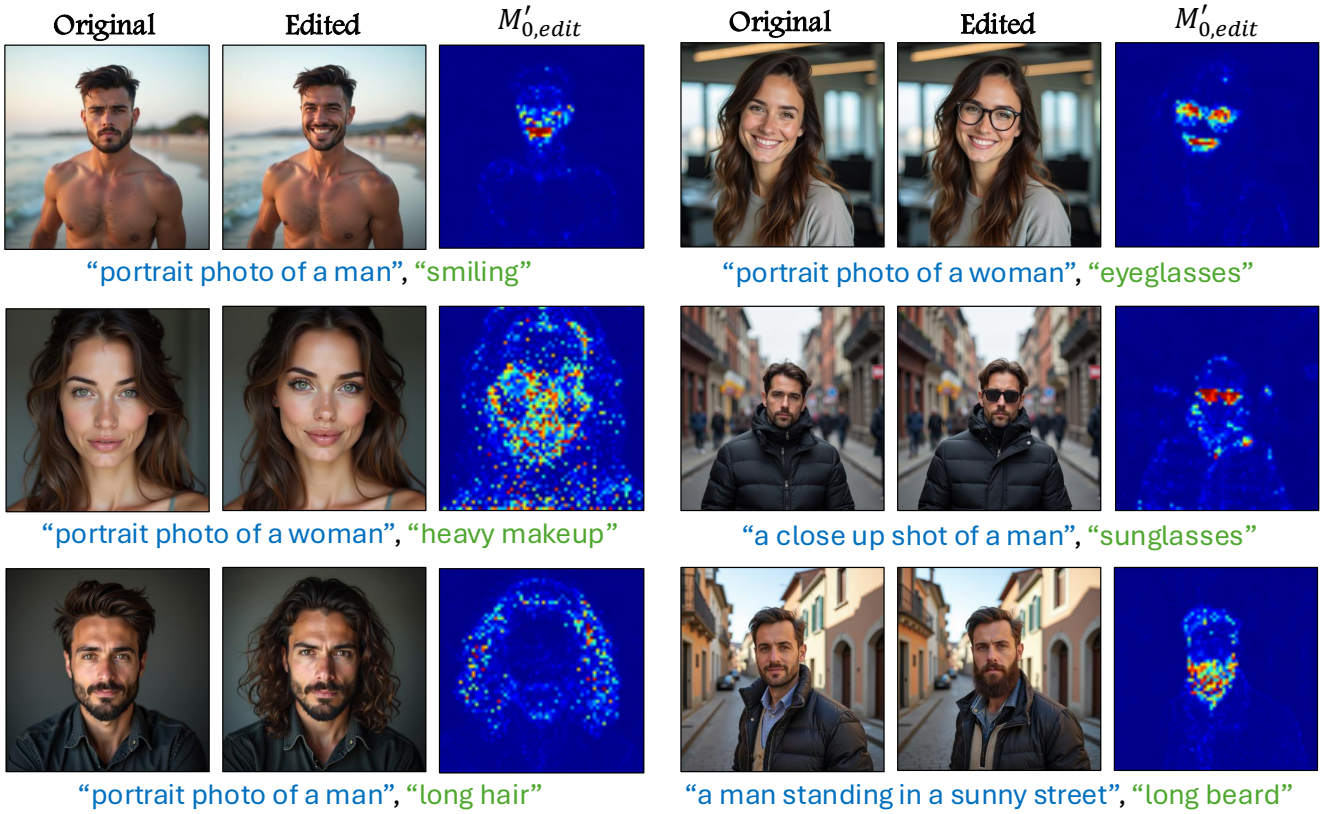

Figure 13. **Visualization of Attention Masks.** We provide visualizations of the thresholding masks $M'_{i,edit}$, to highlight the ability of targeting semantics described by the editing conditions c_e. In all of the examples we provide, the attention masks corresponding to the first text token is visualized, as addressed in our methodology section. As our visualizations show, the masking strategy in *FluxSpace* can successfully address the related semantics, to constrain the edits only with the relevant regions.

Figure 14. **Supplementary Ablations on the Masking Threshold τ_m.** We present additional ablations on the masking threshold τ_m. In addition to ablations provided on face editing in Fig. 6, we provide examples on different domains to show the generalizability if our method on different imaging settings. We provide the conditions for each example as generation condition, editing condition, respectively.

Figure 15. **Supplementary Ablations on the Editing Timesteps.** We provide supplementary ablations on the impact of the timestep selected as the start of the editing process. For each example, we start the editing iterations on the selected timestep t. Even though performing the edit succeeds in reflecting the edit semantic, constraining the timesteps improve the disentanglement capabilities of *FluxSpace*.

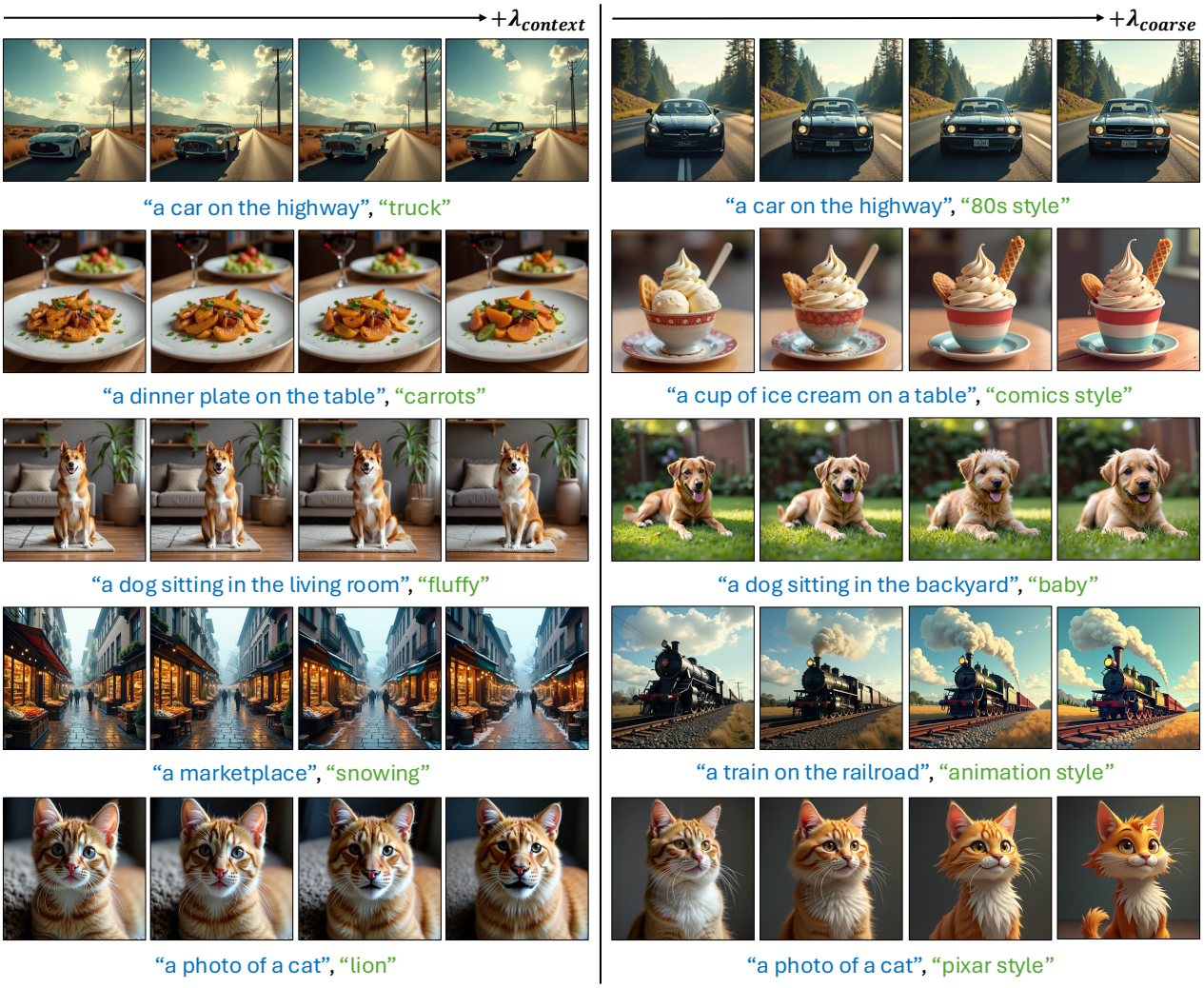

Figure 16. **Supplementary Ablations on λ_{coarse} and λ_{fine}.** We provide supplementary ablations on coarse and fine-grained editing scales λ_{coarse} and λ_{fine} on non-human subjects. As we demonstrate with the provided examples, *FluxSpace* offers control over the editing scale for both coarse level and fine-grained edits. We provide the generation and edit conditions below each example as a reference.