SliderEdit: Continuous Image Editing with Fine-Grained Instruction Control

Supplementary Material

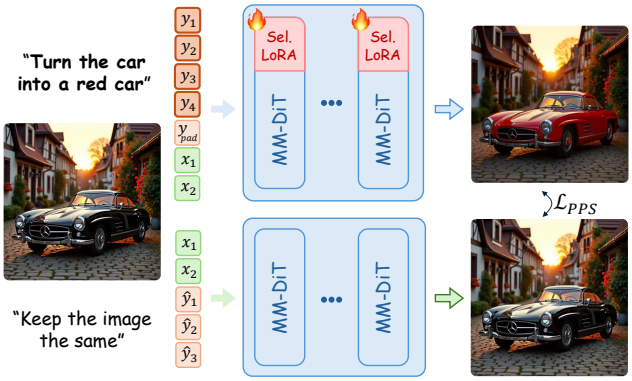

Figure 8. **Simplified Partial Prompt Suppression (SPPS).** SPPS applies the same suppression objective as PPS but treats the entire edit prompt as a single instruction. During training, a second (bottom-row) forward pass is performed to obtain a neutralized image—either using an empty prompt ("") or a neutral textual instruction (e.g., "keep the image the same"). This simple formulation effectively teaches the adapter to suppress undesired edit effects and generalizes well to multi-instruction editing scenarios.

6. Related Works

6.1. Diffusion Models and Flow Matching

Diffusion models belong to a class of generative models based on stochastic differential equations (SDE). The core idea is to gradually corrupt data by adding noise through a stochastic forward process until the original data distribution becomes a simple Gaussian distribution. This process can be described as:

$$dx = f(x,t)dt + g(t)dW_t,$$

where $f(x,t)$ denotes the drift term, $g(t)$ represents the diffusion coefficient, and dW_t is the Wiener process (an infinitesimal step of Brownian motion, representing a small random Gaussian perturbation). The model then learns the reverse process, which reconstructs the original data distribution from pure noise. Mathematically, this reverse-time SDE is written as:

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dW_t,$$

where $\nabla_x \log p_t(x)$ is the score function, representing the gradient of the log-density of the data distribution at time t. Intuitively, the score function tells the model in which direction to move each noisy sample to recover the data distribution. In practice, diffusion models are trained to approximate this score function using a neural network $s_\theta(x,t)$. Training minimizes the score matching loss, defined as:

$$\mathbb{E}_{t\sim U(0,T), x\sim p_t(x)}[\lambda(t)||\nabla_x \log p_t(x) - s_\theta(x,t)||^2],$$

Figure 9. Qualitative results of GSTLoRA on text editing.

where $\lambda(t)$ is a time-dependent weighting function. Once trained, the model can sample new data by simulating the learned reverse process starting from Gaussian noise.

Flow matching methods are closely related to diffusion models, designed for training Continuous Normalizing Flows. The key idea is to learn a deterministic transformation that maps an initial noise distribution to the target data distribution by integrating an ordinary differential equation (ODE). The evolution of a sample x over time is governed by a time-dependent vector field $v_\theta(x,t)$, defined as:

$$\frac{dx}{dt} = v_\theta(x,t),$$

where $v_\theta(x,t)$ is a neural network parameterizing the vector field to be learned. Training involves aligning this learned field with a predefined target vector field $v_t(x)$, which describes how samples should flow from noise to data at each

time step. This is achieved by minimizing the flow matching loss:

$$\mathbb{E}_{t\sim U(0,T),x\sim p_t(x)}[|v_\theta(x,t) - v_t(x)|^2],$$

where $p_t(x)$ represents intermediate distributions along the transformation path from the initial to the final data distribution. Unlike diffusion models, which rely on stochastic SDE trajectories involving random noise, flow matching employs deterministic ODE trajectories. This eliminates the stochasticity in sampling and generally leads to faster and more efficient training and inference. As a result, flow matching can be viewed as a computationally efficient deterministic counterpart to diffusion models.

7. SliderEdit: Continuous Image Editing

7.1. Simplified Partial Prompt Suppression Loss

While the main Partial Prompt Suppression (PPS) objective requires selectivel suppressing an individual instruction $\mathcal{P}_i$ within a composite prompt $\mathcal{P}$, the Simplified PPS (SPPS) variant adopts a more streamlined approach that reduces this complexity a bit while maintaining strong generalization.

In SPPS, each training sample is treated as a single-instruction editing instance, i.e., $\mathcal{P} = \{\mathcal{P}_1\}$. The model learns to suppress the visual influence of this sole instruction by minimizing the difference between the denoising prediction of the adapted model when conditioned on $\mathcal{P}_1$ and that of the frozen base model when the prompt is removed entirely (or replaced with a prompt that acts as a null instruction, e.g., "keep the image the same"). Formally, the loss follows the same structure as $\mathcal{L}_{\texttt{PPS}}$:

$$\mathcal{L}_{\texttt{SPPS}} = \|\epsilon_{M_\theta(\mathcal{P}_1)}(Z, X_{\text{orig}}, \mathcal{P}_1) - \epsilon(Z, X_{\text{orig}}, \varnothing\})\|$$

This encourages the adapter to learn how to neutralize the edit induced by $\mathcal{P}_1$, thereby isolating its corresponding representation within the model. Figure 8 visualizes the SPPS training pipeline.

Despite its simplicity, SPPS offers several practical advantages. It removes the need to parse multi-instruction prompts or identify token-level boundaries between sub-instructions, allowing efficient training on general instruction-based editing datasets, including those containing only single-instruction pairs. Moreover, the adapters trained with SPPS exhibit strong robustness and compositional generalization, performing effectively even when applied to multi-instruction edits at inference time. However, PPS provides finer-grained supervision, leading to more disentangled and well-localized adaptations across different instruction dimensions, which results in better control when handling complex multi-instruction edits.

8. Experiments

8.1. Implementation Details

We use FLUX-Kontext and Qwen-Image-Edit[1] as our base models. All models are trained with the ℓ_{SPPS} loss, chosen for its simplicity, efficiency, and strong generalization. We observe that $\mathcal{L}_{\text{PPS}}$ provides more robust and disentangled control for multi-instruction setups when used with STLoRA (see Appendix 8.3.1). Training is performed on a small subset (1k–8k samples) of the GPT-Image-Edit-1.5M dataset [41]. For STLoRA, where we train a token-aware adapter [39], we train both base models for 1,000 iterations with a batch size of 8, observing early convergence around iterations 400–500 but continuing to 1,000 for consistency. For GSTLoRA, we train FLUX-Kontext for 300 iterations with a batch size of 4. We employ the AdamW optimizer with a learning rate of 1×10^{-4}, no warm-up, and train across all diffusion timesteps. All experiments are conducted on a single NVIDIA H100-SXM GPU using mixed-precision (`bfloat16`) training with gradient checkpointing for memory efficiency. The LoRA modules have a rank of 16 and zero dropout, and are applied to the Q, K, V, and output projections of the attention layers, as well as to the two additional linear projections in each transformer block. These settings provide a stable and memory-efficient training setup, enabling rapid convergence across all models. Overall, our training is computationally highly lightweight and data-efficient. Furthermore, consistent with prior observations in [9, 46], we found that training adapters on only a subset of transformer blocks can achieve performance comparable to training all blocks. Also, following insights from [44, 45], applying adapters at every denoising timestep may not be necessary for effective editing. We leave a comprehensive investigation of these efficiency-oriented design choices for future work.

8.2. Quantitative Results

8.2.1. Metrics

Continuity. Given a sequence of similarity scores $\{s_1, \ldots, s_\delta\}$ corresponding to increasing α values, we expect these scores to change smoothly and approximately uniformly between $\min(s_i)$ and $\max(s_i)$. To quantify this, we compute a chi-squared statistic,

$$\chi^2 = \sum_{i=1}^{\delta} \frac{(O_i - E)^2}{E},$$

where O_i denotes the observed count in each bin (number of similarity scores s_j falling within the i-th bin), and E denotes the expected count per bin under a uniform distribution ($E = 1$). We report $(\chi^2_{\text{agg}}/\text{dof})^{-1}$ (dof :degrees of

[1]We adopt Qwen-Image-Edit-2509, an updated version with improved performance and stronger identity preservation.

Figure 10. **Qualitative results of STLoRA on a 3-instruction edit.** The model demonstrates smooth and continuous control over the strength of each instruction in a disentangled manner.

Figure 11. Qualitative results of STLoRA on a 2-instruction edit for text editing.

freedom) as our continuity metric—larger values indicate higher continuity and smoother edit trajectories. For 2D and 3D edit spaces, we apply an analogous chi-squared test to evaluate the uniformity of the sample distribution across the corresponding grids.

Disentanglement. To evaluate disentanglement, we measure how well the model isolates the intended edit without affecting unrelated aspects, such as identity or background. First, we assess *identity preservation* using cosine distance in the identity embedding space obtained from *ArcFace* [8], where lower distances indicate stronger identity consistency. To capture more general visual changes, we compute feature distances between edited images $\mathcal{I}_i$ with the origin image using multiple perceptual metrics: *LPIPS* [48] (using both AlexNet [23] and VGG [37] backbones) and *DINOv2* [5, 32]. While LPIPS focuses on low-level perceptual similarity, DINO captures higher-level semantic consistency, allowing us to evaluate both appearance-level and structural disentanglement.

8.2.2. Baselines

We consider different baselines depending on the number of edit instructions γ used in the prompt.

For the case of a single-instruction setting ($\gamma = 1$), we compare *GSTLoRA* (Ours) and *STLoRA* (Ours) with *Explicit CFG* and *Implicit CFG*, all implemented on top of the *FLUX-Kontext* model, as well as Concept-Slider [10] and Continuous Attribute Control [3]. Implicit CFG refers to the classifier-free guidance (CFG) mechanism applied in an implicit manner. FLUX-Kontext is a *guidance-distilled* model, meaning that at inference time it does not explicitly

Figure 12. **Qualitative Comparison between PPS and SPPS.** PPS produces a more disentangled and smoother interpolation space in multi-instruction editing scenarios, offering finer control over individual instruction directions compared to SPPS.

perform CFG as:

$$\epsilon_{\text{CFG}} = \epsilon_{\text{uncond}} + s\big(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}\big),$$

where ϵ_{cond} and ϵ_{uncond} denote the conditional and unconditional predictions, respectively, and s is the guidance scale. Instead, the model internally learns to approximate the effect of a given s, allowing us to vary this parameter to implicitly control guidance strength. However, as observed in our experiments, this implicit scaling provides only limited control over the edit intensity.

To enable explicit guidance, we first set the model's internal (implicit) guidance scale to $s = 1$, effectively recovering the base (unguided) model. We then apply explicit CFG during inference using $\epsilon' = \epsilon_{\text{uncond}} + w\big(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}\big)$, where w is the external CFG scale. This requires two forward passes through the model—one with the conditioning prompt and one without".

Concept-Slider and Continuous Attribute Control enable fine-grained attribute manipulation in text-to-image models. While they can be adapted to image editing via inversion methods [12, 30], their performance in this setting is comparatively limited.

For cases involving multiple edit instructions ($\gamma > 1$), Explicit CFG, Implicit CFG, and GSTLoRA cannot independently control individual edit directions. This limitation highlights the advantage of *STLoRA*, which enables disentangled, per-instruction control in multi-instruction editing scenarios. As Concept-Slider and Continuous Attribute Control show limited effectiveness even for single-instruction edits ($\gamma = 1$), we omit them from this setting. We evaluate STLoRA using both *FLUX-Kontext* and *Qwen-Image-Edit* models.

8.3. Qualitative Results

We provide additional qualitative results to further illustrate the capabilities of *SliderEdit* and its variants across a diverse range of editing tasks.

Figure 14 showcases diverse examples generated using *GSTLoRA*, demonstrating smooth and continuous control over both local and global edits. The model effectively interpolates between different edit strengths, producing coherent intermediate images without abrupt transitions.

To further evaluate its capability in fine-grained manipulation, Figures 15 and 16 present qualitative results on face-editing tasks. The model can accurately and continuously adjust facial attributes such as hair length, curliness, makeup, skin tone, hair color, and age, as well as facial expressions including smiling, anger, and surprise. In addition, Figure 9 demonstrates GSTLoRA's versatility in *text editing*. The model enables continuous adjustment of textual attributes such as font color, style, and weight.

Figures 17 and 10 illustrate qualitative results of *STLoRA* on multi-instruction editing tasks. In the 2-instruction setting, the model produces a smooth and interpretable 2D in-

Figure 13. **Qualitative Comparison with Baselines.** While SliderEdit (GSTLoRA variant here) and Explicit Guidance produce high-quality edits, Concept-Slider and Continuous Attribute Control perform poorly on real image editing, as they are primarily designed for text-to-image generation and rely on indirect inversion-based adaptation.

terpolation space, where each axis corresponds to a distinct instruction direction. Extending this to 3-instruction scenarios (Figure 10), STLoRA maintains disentangled control, allowing continuous modulation of each instruction independently. We further demonstrate STLoRA's capability on *text editing* tasks in Figure 11, where the model learns disentangled control over multiple text attributes (e.g., font style and color).

Overall, these results highlight the flexibility and generality of the proposed framework across domains, showing that both GSTLoRA and STLoRA enable smooth, continuous, and disentangled control over diverse editing operations.

8.3.1. PPS vs SPPS

We compare the Partial Prompt Suppression (PPS) and Simplified PPS (SPPS) objectives to assess their effect on disentanglement and control quality. As illustrated in Figure 12, both objectives enable smooth and continuous interpolation along edit directions. However, PPS produces a more disentangled latent space, allowing finer and more independent control over each instruction, while SPPS serves as a simpler yet effective alternative that achieves comparable results in most cases.

Some degree of attribute entanglement persists across all models, including the underlying base model. For instance, even when using the base instruction-based editing model, modifying a person's skin tone can unintentionally affect correlated features such as hair color or lighting. This behavior arises from inherent attribute coupling in the generative model itself, rather than from limitations introduced by our sliders.

8.3.2. Comparison with other baselines

As shown quantitatively in Table 1 and discussed in Section 4, Concept-Slider and Continuous Attribute Control perform poorly on real image editing tasks due to their indirect adaptation from text-to-image generation. Here, we provide qualitative examples in Figure 13 for visual comparison. While SliderEdit (GSTLoRA variant in this case) and Explicit Guidance produce smooth, coherent, and faithful edits aligned with the input instructions, Concept-Slider and Continuous Attribute Control often fail to maintain image fidelity or accurately follow the target modification. These qualitative results further confirm the quantitative findings, demonstrating that SliderEdit enables both fine-grained control and high-quality real image editing.

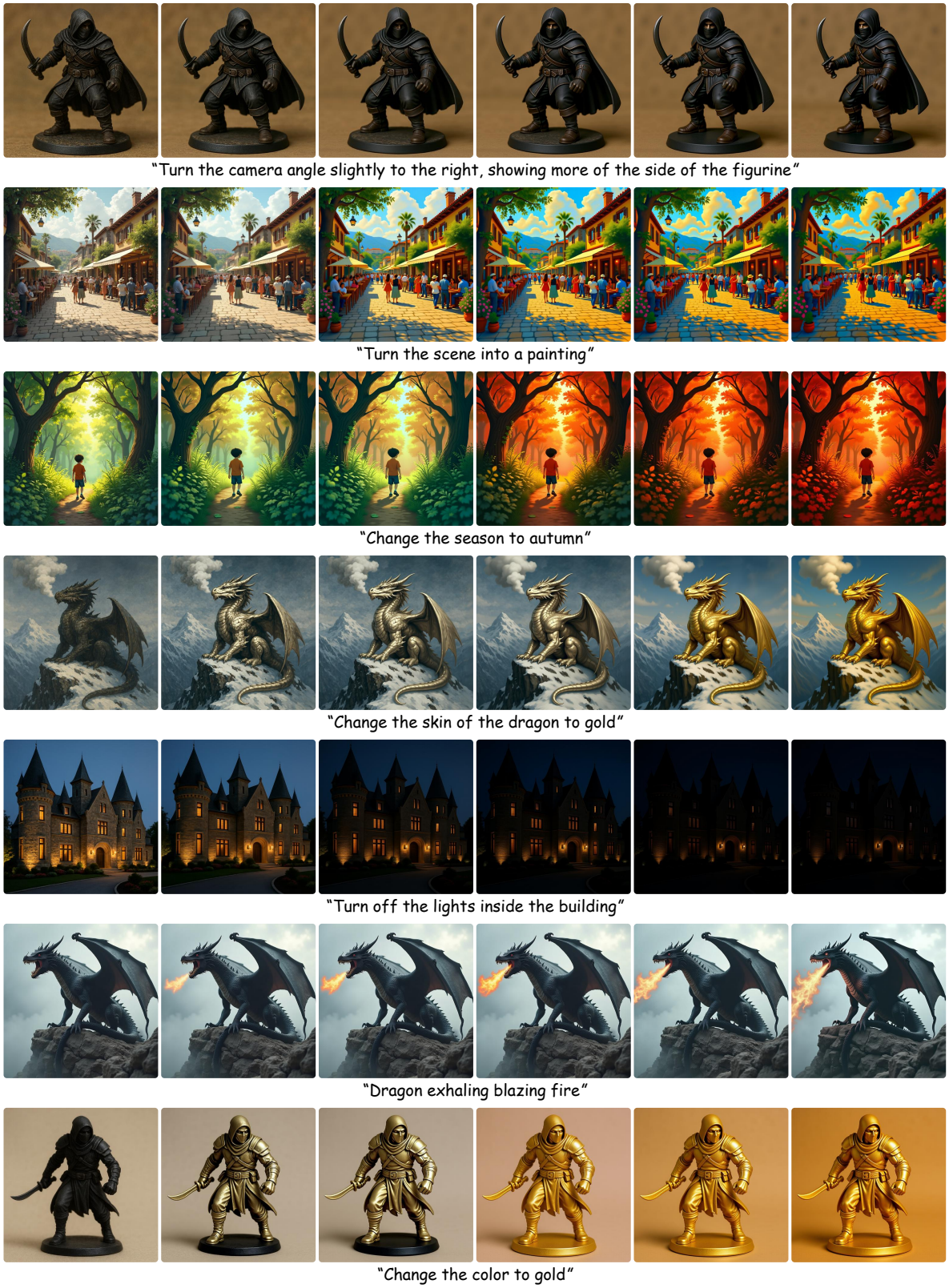

Figure 14. **Qualitative Samples of GSTLoRA.** The model emonstrates smooth, continuous control over the strength of both local and global edits.

Figure 15. Qualitative results of GSTLoRA on face editing

Figure 16. Qualitative results of GSTLoRA on face editing

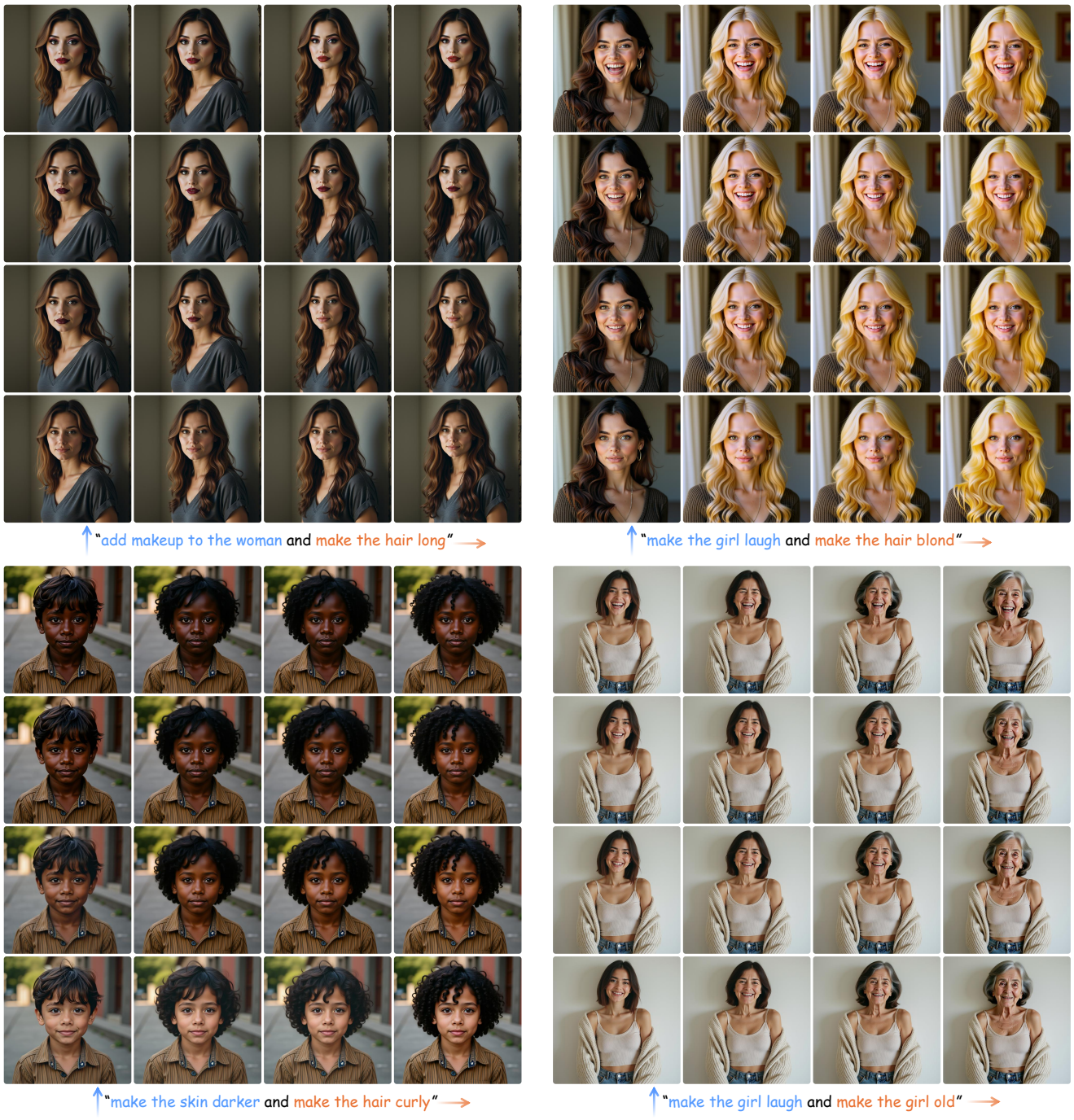

Figure 17. **Qualitative results of STLoRA on 2-instruction edits.** The model demonstrates smooth, continuous control over the strength of both directions.