Stable Flow: Vital Layers for Training-Free Image Editing

Omri Avrahami[1,2] Or Patashnik[1,3] Ohad Fried[4] Egor Nemchinov[1]
Kfir Aberman[1] Dani Lischinski[2] Daniel Cohen-Or[1,3]

[1]Snap Research [2]The Hebrew University of Jerusalem [3]Tel Aviv University [4]Reichman University

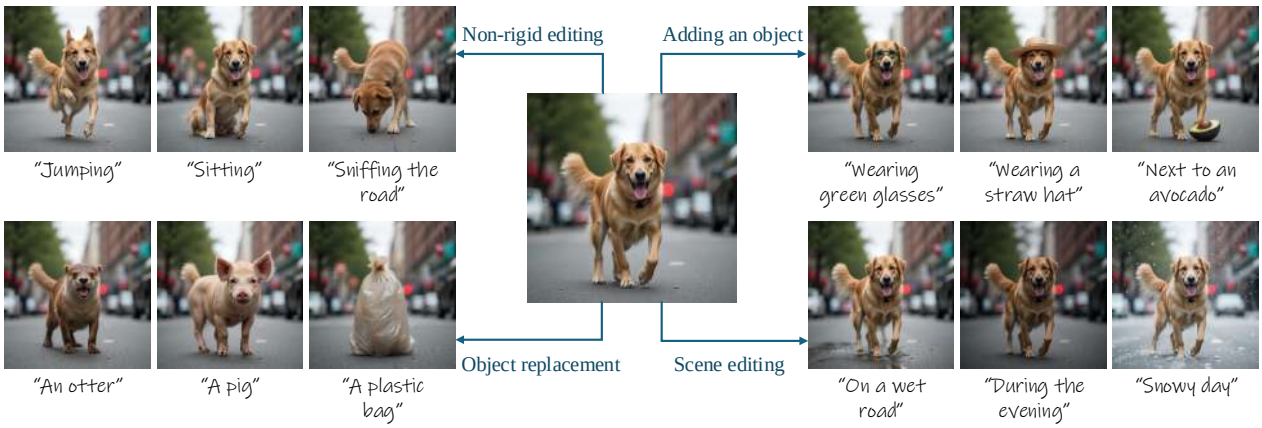


Figure 1. **Stable Flow.** Our training-free editing method is able to perform *various* types of image editing operations, including non-rigid editing, object addition, object removal, and global scene editing. These different edits are done using the *same* mechanism.

Abstract

Diffusion models have revolutionized the field of content synthesis and editing. Recent models have replaced the traditional UNet architecture with the Diffusion Transformer (DiT), and employed flow-matching for improved training and sampling. However, they exhibit limited generation diversity. In this work, we leverage this limitation to perform consistent image edits via selective injection of attention features. The main challenge is that, unlike the UNet-based models, DiT lacks a coarse-to-fine synthesis structure, making it unclear in which layers to perform the injection. Therefore, we propose an automatic method to identify "vital layers" within DiT, crucial for image formation, and demonstrate how these layers facilitate a range of controlled stable *edits, from non-rigid modifications to object addition, using the* same *mechanism. Next, to enable real-image editing, we introduce an improved image inversion method for flow models. Finally, we evaluate our approach through qualitative and quantitative comparisons, along with a user study, and demonstrate its effectiveness across multiple applications.*

1. Introduction

Over the recent years, we have witnessed an unprecedented explosion in creative applications of generative models, fueled by diffusion-based models [36, 78–80]. Recent models, such as FLUX [46] and SD3 [25], have replaced the traditional UNet architecture [73] with the Diffusion Transformer (DiT) [63], and adopted flow matching [5, 49, 50] as a superior alternative for training and sampling.

These flow-based models are based on optimal transport conditional probability paths, resulting in faster training and sampling, compared to diffusion models. This is attributed [49] to the fact that they follow straight line trajectories, rather than curved paths. One of the known consequences of this difference, however, is that these models exhibit lower diversity than previous diffusion models [26], as shown in Figure 2(1-2). While reduced diversity is generally considered an undesirable characteristic, in this paper, we suggest leveraging it for the task of training-free image editing, as shown in Figure 2(3) and Figure 1.

Specifically, we explore image editing via parallel generation [4, 18, 91], where features from the generative trajectory of the source (reference) image are injected into the trajectory of the edited image. Such an approach has been shown effective in the context of convolutional UNet-based

Project page is available at: https://omriavrahami.com/stable-flow

This research was performed while Omri was at Snap.

Figure 2. **Leveraging Reduced Diversity.** Using the same initial seed with different editing prompts, diffusion models such as (1) SDXL generate diverse results (different identities of the dog and the cat), while (2) FLUX generates a more stable (less diverse) set of results out-of-the-box. However, there are still some unintended differences (the dog is standing in the leftmost column and sitting in the others, the color of the cat is changing, and the road is different on the right). Using our approach, (3) Stable Flow, the edits are stable, maintaining consistency of the unrelated content.

diffusion models [18], where the roles of the different attention layers are well understood. However, such understanding has not yet emerged for DiT [63]. Specifically, DiT does not exhibit the same fine-coarse-fine structure of the UNet [63], hence it is not clear which layers should be tampered with to achieve the desired editing behavior.

To address this gap, we analyze the importance of the different components in the DiT architecture, in order to determine the subset that should be injected while editing. More specifically, we introduce an *automatic* method for detecting a set of *vital layers* — layers that are essential for the image formation — by measuring the deviation in image content resulting from bypassing each layer. We show that there is no simple relationship between the vitality of a layer and its position in the architecture, *i.e.*, the vital layers are spread across the transformer.

A close examination of the vital layers suggests that the injection of features into these layers strikes a good balance in the multimodal attention between the reference image content and the editing prompt. Consequently, limiting the injection of features to *only* the vital layers tends to yield a stable edit, *i.e.*, an edit that changes only the part(s) specified by the text prompt, while leaving the rest of the image intact, as demonstrated in Figure 2(3). We demonstrate that performing *the same* feature injection, enables performing a *variety* of image edits, including non-rigid editing, addition of objects, and scene changes, as demonstrated in Figure 1.

In order to support editing real images, it is typically necessary to invert them first [53, 79]. We employ an Inverse Euler Ordinary Differential Equation (ODE) solver to invert real images in the FLUX model [46]. However, this method fails to reconstruct the input image in a satisfactory manner. To improve reconstruction accuracy, we introduce an out-of-distribution (OOD) nudging technique, in which we apply a small scalar perturbation to the clean latent before inverting it. We demonstrate that this makes FLUX less prone to undesired changes in the image during the forward pass.

Finally, we compare our method against its baselines qualitatively and quantitatively, and reaffirm the results with a user study. We also demonstrate several applications of our method.

In summary, our contributions are: (1) we propose an automatic method to detect the set of vital layers in DiT models and demonstrate how to use them for image editing; (2) we are the first method to harness the limited diversity of flow-based models to perform *different* image editing tasks using the *same* mechanism; and (3) we present an extension that allows editing real images using the FLUX model.

2. Related Work

Text-Driven Image Editing. After the emergence of text-to-image models, many works have suggested using them for various applications [7, 12, 20, 37, 75], including text-driven image editing tasks [39, 65]. SDEdit [52] addressed the image-to-image translation task adding noise and denoising with a new prompt. Blended Diffusion [8, 9] suggested performing localized image editing [15, 38, 47, 55, 62] incorporating an input mask into the diffusion process in a training-free manner, while GLIDE [54], ImagenEditor [89], and SmartBrush [93] offered fine-tuning the model on a given input mask. Other works suggested inferring the mask from the input image [21, 88] or via an additional click input [69]. Other works [18, 34, 60, 84] offered to inject information from the input image using parallel generation. While some methods [13, 45, 85] suggested fine-tuning the model per-image, a recent line of work suggested training a designated model on a large synthetic dataset for instruction-based edits [17, 76, 97]. However, none of the above methods is a training-free method that supports non-rigid editing, object adding/replacement and scene editing altogether. Concurrency, Add-it [81] offers a solution to the task of object addition using FLUX. Our method, on the other hand, supports other types of image editing (*e.g.*, non-rigid editing, object replacement).

Image Inversion. In the realm of generative models, *inversion* [92] is the task of finding a code within the latent space of a generator [31, 43, 44] that faithfully reconstructs a given image. Initial methods were developed for GAN

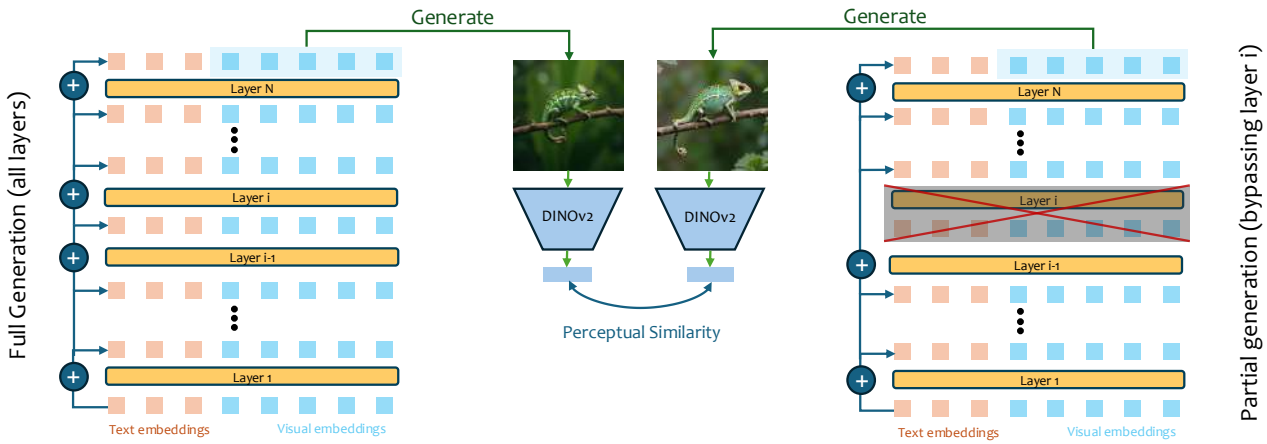


Figure 3. **Layer Removal.** (Left) Text-to-image DiT models consist of consecutive layers connected through residual connections [33]. Each layer implements a multimodal diffusion transformer block [25] that processes a combined sequence of text and image embeddings. (Right) For each DiT layer, we performe an ablation by bypassing the layer using its residual connection. Then, we compare the generated result on the ablated model with the complete model using a perceptual similarity metric.

models [1–3, 14, 24, 59, 64, 70, 71, 83, 95, 99, 100], and more recently for diffusion-based models [16, 23, 29, 32, 40, 51, 53, 58, 79, 87]. In this work, we suggest inverting a real image in flow models using latent nudging, as we found that the standard inverse ODE solver is insufficient.

3. Method

Our goal is to edit images based on text prompts while faithfully preserving the unedited regions of the source image. Given an input image x and an editing prompt p, we aim to generate a modified image $\hat{x}$ that exhibits the desired changes specified by p while maintaining the original content elsewhere. We leverage the limited diversity of the FLUX model and further constrain it to enable such *stable* image edits through selective injection of attention features of the source image into the process that generates $\hat{x}$. Our approach is described in more detail below. In Section 3.1, we evaluate layer importance in the DiT model by analyzing the perceptual impact of layer removal (Figure 3). Next, in Section 3.2, we employ the most influential layers (termed *vital* layers) for image editing through attention injection. Finally, in Section 3.3, we extend our method to real image editing by inverting images into the latent space using the Euler inverse ODE solver, enhanced by *latent nudging*.

3.1. Measuring the Importance of DiT Layers

Recent text-to-image diffusion models [36, 66, 68, 72, 74] predominantly use CNN-based UNets, which exhibit well-understood layer roles. In discriminative tasks, early layers detect simple features like edges, while deeper layers capture higher-level semantic concepts [77, 96]. Similarly, in generative models, early-middle layers determine shape and color, while deeper layers control finer details [43]. This structure has been successfully exploited in text-driven editing [18, 30, 84] through targeted manipulation of UNet decoder layers [73].

In contrast, state-of-the-art text-to-image DiT [63] models (FLUX [46] and SD3 [25]) employ a fundamentally different architecture, as shown in Figure 3(left). These models consist of consecutive layers connected through residual connections [33], without convolutions. Each layer implements a multimodal diffusion transformer block [25] (MM-DiT-Block) that processes a combined sequence of text and image embeddings. Unlike in UNets, the roles of the different layers are not yet intuitively clear, making it challenging to determine which layers are best suited for image editing.

To quantify layer importance in the FLUX model, we devised a systematic evaluation approach. Using ChatGPT [56], we automatically generated a set P of $k = 64$ diverse text prompts, and draw a set S of random seeds. Each of these prompts was used to generate a reference image, yielding in a set G_{ref}. For each DiT layer $\ell \in \mathbb{L}$, we performed an ablation by bypassing the layer using its residual connection, as illustrated in Figure 3(right). This process generated a set of images G_ℓ from the same prompts and seeds. See the supplementary material for more details.

To assess the impact of each layer, we measured the perceptual similarity between G_{ref} and G_ℓ and using DINOv2 [57] (see Figure 3). The results, plotted in Figure 4, show that removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. We formally define the *vitality* of layer ℓ as:

$$vitality(\ell) = 1 - \frac{1}{k} \sum_{s \in S, p \in P} d(M_{\text{full}}(s,p), M_{\text{-}\ell}(s,p)), \quad (1)$$

where M_{full} represents the complete model, $M_{\text{-}\ell}$ denotes the

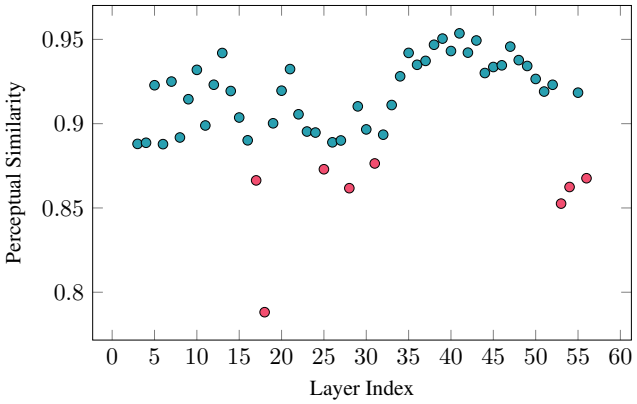


Figure 4. **Layer Removal Quantitative Comparison.** As explained in Section 3.1, we measured the effect of removing each layer of the model by calculating the perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images (Figure 5). As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

model with layer ℓ omitted, and $d(\cdot,\cdot)$ is the perceptual similarity metric. The set of vital layers V is then defined as:

$$V = \{\ell \in \mathbb{L} \mid \mathit{vitality}(\ell) \geq \tau_{vit}\}\,, \tag{2}$$

where τ_{vit} is the vitality threshold.

Figure 5 illustrates the qualitative differences between vital and non-vital layers. While bypassing non-vital layers results in minor alterations, removing vital layers leads to significant changes: complete noise generation (G_0), global structure and identity changes (G_{18}), and alterations in texture and fine details (G_{56}).

3.2. Image Editing using Vital Layers

Given a source image x generated with a known seed s and prompt p, we aim to modify p and generate an edited image $\hat{x}$ that exhibits the desired changes, while otherwise preserving the source content. We adapt the self-attention injection mechanism, previously shown effective for image and video editing [18, 91] in UNet-based diffusion models, to the DiT-based FLUX architecture. Since each DiT layer processes a sequence of image and text embeddings, we propose generating both x and $\hat{x}$ in parallel while *selectively replacing* the image embeddings of $\hat{x}$ with those of x, but only within the vital layers set V.

Remarkably, as shown in Figure 1, this training-free approach successfully performs diverse editing tasks, including non-rigid deformations, object addition, object replacement, and global modifications, all using the *same* set of vital layers V.

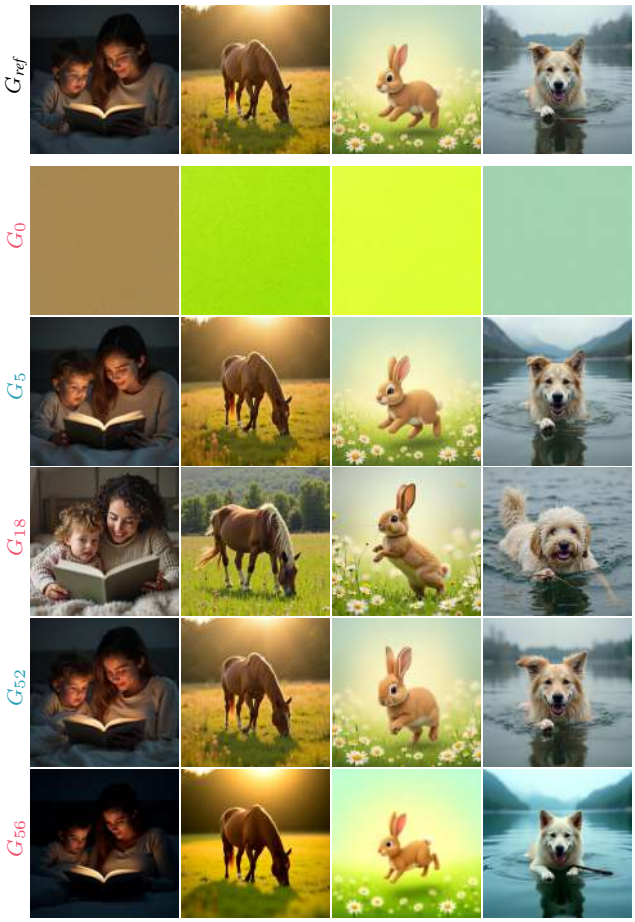


Figure 5. **Layer Removal Qualitative Comparison.** As explained in Section 3.1, we illustrate the qualitative differences between vital and non-vital layers. While bypassing non-vital layers (G_5 and G_{52}) results in minor alterations, bypassing vital layers leads to significant changes: complete noise generation (G_0), global structure and identity changes (G_{18}), and alterations in texture and fine details (G_{56}).

To understand this effectiveness, we analyze the multimodal attention patterns in FLUX. Each visual token simultaneously attends to all visual and text tokens, with attention weights normalized across both modalities. Figure 6 contrasts the attention patterns in vital versus non-vital layers at two key points: a yellow point in a region that should remain unchanged (requiring copying from the reference image), and a red point in an area targeted for editing (requiring generation based on the text prompt). In vital layers (left), points meant to remain unchanged show dominant attention to visual features, while points targeted for editing exhibit stronger attention to relevant text tokens (e.g., *"avocado"*). Conversely, non-vital layers (right) show predominantly image-based attention even in regions marked for editing. This suggests that injecting features into vital layers strikes a good multimodal attention balance between preserving source content and incorporating text edits.

Figure 6. **Multi-Modal Attention Distribution.** Given an input image of a man, we edit it to hold an avocado by injecting the reference image in the vital layers only (left) or in the non-vital layers (right), and visualize the multimodal attention of two points: a yellow point in a region that should remain unchanged (requiring copying from the reference image), and a red point in an area targeted for editing (requiring generation based on the text prompt). As can be seen, in vital layers (left), points meant to remain unchanged show dominant attention to visual features, while points targeted for editing exhibit stronger attention to relevant text tokens (e.g., *"avocado"*). Conversely, non-vital layers (right) show predominantly image-based attention even in regions marked for editing. This suggests that injecting features into vital layers strikes a good multimodal attention balance between preserving source content and incorporating text-guided modifications.

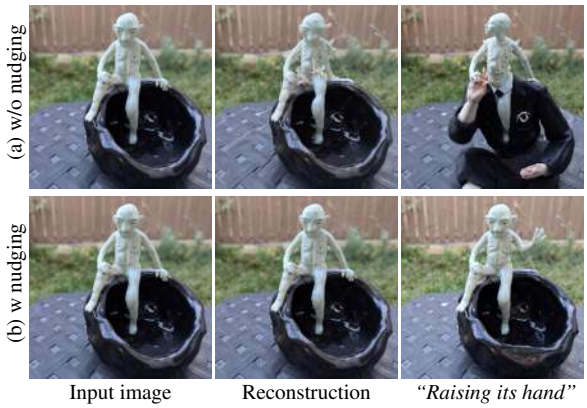


Figure 7. **Latent Nudging.** As described in Section 3.3, when inverting a real image, (a) a simple inverse Euler ODE solver leads to corrupted image reconstructions and unintended modifications during editing. On the other hand, (b) using our latent nudging technique significantly reduces reconstruction errors and better constrains edits to the intended regions.

3.3. Latent Nudging for Real Image Editing

Flow models generate samples by matching a prior distribution p_0 (Gaussian noise) to a data distribution p_1 (the image manifold). In the space $\mathbb{R}^d$, we define two key components: a probability density path $p_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}_{>0}$, which specifies time-dependent probability density functions ($\int p_t(x)dx = 1$), and a vector field $u_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. This vector field generates a flow $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ through the ordinary differential equation (ODE): $\frac{d}{dt}\phi_t(x) = u_t(\phi_t(x)); \phi_0(x) = x$. Transforming a sample from p_0 to a sample in p_1 is achieved using ODE solvers such as Euler.

To edit real images, we must first *invert* them into the latent space, transforming samples from p_1 to p_0. We initially implemented an inverse Euler ODE solver for FLUX by reversing the vector field prediction. Given the forward Euler step:

$$z_{t-1} = z_t + (\sigma_{t+1} - \sigma_t) * u_t(z_t) \tag{3}$$

where z_t represents the latent at timestep t, σ_t is the optimal transport standard deviation at time t, and u_t is the learned vector field, we proposed the inverse step:

$$z_t = z_{t-1} + (\sigma_t - \sigma_{t+1}) * u_t(z_{t-1}) \tag{4}$$

assuming $u_t(z_t) \approx u_t(z_{t-1})$ for small steps.

However, as Figure 7(a) demonstrates, this approach proves insufficient for FLUX, resulting in corrupted image reconstructions and unintended modifications during editing. We hypothesize that the assumption $u(z_t) \approx u(z_{t-1})$ does not hold, which causes the model to significantly alter the image during the forward process. To address this, we introduce *latent nudging*: multiplying the initial latent z_0 by a small scalar $\lambda = 1.15$ to slightly offset it from the training distribution. While this modification is visually imperceptible (Figure 7(b)), it significantly reduces reconstruction errors and constrains edits to the intended regions. See the supplementary material for more details

4. Experiments

In Section 4.1 we compare our method against its baselines, both qualitatively and quantitatively. Next, in Section 4.2 we conduct a user study and report results. Furthermore, in Section 4.3 we present the ablation study results. Finally, in Section 4.4 we demonstrate several applications.

Table 1. **Quantitative Comparison.** We compare our method against the baselines in terms of text similarity (CLIP_{txt}), image similarity (CLIP_{img}) and image-text direction similarity (CLIP_{dir}). As can be seen, P2P+NTI [34, 53], Instruct-P2P [17], and MasaCTRL [18] suffer from low similarity to the text prompt. SDEdit [94] and MagicBrush [97] adhere more to the text prompt, but they struggle with image similarity and image-text direction similarity. Our method, on the other hand, achieves better image and image-text direction similarity.

Method	CLIP_{txt} (↑)	CLIP_{img} (↑)	CLIP_{dir} (↑)
SDEdit [94]	**0.24**	0.71	0.07
P2P+NTI [34, 53]	0.21	0.76	0.08
Instruct-P2P [17]	0.22	0.87	0.07
MagicBrush [97]	**0.24**	0.88	0.11
MasaCTRL [18]	0.20	0.76	0.03
Stable Flow (ours)	0.23	**0.92**	**0.14**

Table 2. **Ablation Study.** We conduct an ablation study and find that performing attention injection in all the layers or performing an attention extension in all the layers significantly reduces the text similarity. Furthermore, performing an attention injection in the non-vital layers or removing the latent nudging reduces the image similarity.

Method	CLIP_{txt} (↑)	CLIP_{img} (↑)	CLIP_{dir} (↑)
Stable Flow (ours)	0.23	0.92	**0.14**
Injection all layers	0.17	**0.98**	0.00
Injection non-vital layers	**0.25**	0.72	0.09
Extension all layers	0.18	**0.98**	0.01
w/o latent nudging	0.22	0.62	0.05

Table 3. **User Study.** We compare our method against the baselines using the standard two-alternative forced-choice format. Users were asked to rate which editing result is better (Ours vs. the baseline) in terms of: (1) target prompt adherence, (2) input image preservation, (3) realism and (4) overall edit quality. We report the win rate of our method compared to each baseline. As shown, our method outperforms the baselines across all categories, achieving a win rate higher than the random chance of 50%.

Ours vs	Prompt Adher. (↑)	Image Pres. (↑)	Realism (↑)	Overall (↑)
SDEdit [52]	69.00%	68.00%	63.66%	70.66%
P2P+NTI [34, 53]	76.00%	71.00%	72.66%	65.33%
Instruct-P2P [17]	76.33%	75.66%	68.00%	60.33%
MagicBrush [97]	61.33%	67.33%	76.66%	74.00%
MasaCTRL [18]	82.33%	80.00%	80.33%	72.00%

4.1. Qualitative and Quantitative Comparison

We compare our method against the most relevant text-driven image editing methods. We re-implement SDEdit [52] using the FLUX.1-dev [46] model, and use the official public implementations of P2P+NTI [34, 53], Instruct-P2P [17], MagicBrush [97] and MasaCTRL [18]. See the supplementary material for more details.

In Figure 8 we compare our method against the baselines qualitatively on real images. As can be seen, SDEdit [52] has difficulty maintaining object identities and backgrounds. P2P+NTI [34, 53] struggles with preserving object identities and with adding new objects. Instruct-P2P [17] and MagicBrush [97] face challenges with non-rigid editing. MasaCTRL [18] struggles with preserving object identities and adding new objects. Our method, on the other hand, is adhering to the editing prompt while preserving the identities.

To quantify the performance of our method and the baselines, we prepared an evaluation dataset based on COCO [48], that in contrast to previous benchmarks [76, 97], also contains non-rigid editing tasks. We start by filtering the dataset automatically to contain at least one prominent non-rigid body. Next, for each image, we use various image editing tasks (non-rigid editing, object addition, object replacement and scene editing) that take into account the prominent object, resulting in a total dataset of 3,200 samples. See the supplementary material for more details.

We evaluated the editing results using three metrics: (1) CLIP_{img} that measures the similarity between the input image and the edited image, (2) CLIP_{txt} that measure the similarity between the edited image and the target editing prompt, and (3) CLIP_{dir} [28, 61] that measures the similarity between the direction of the prompt change and the direction of the image change.

As can be seen in Table 1, P2P+NTI [34, 53], Instruct-P2P [17], and MasaCTRL [18] suffer from low similarity to the text prompt. SDEdit [94] and MagicBrush [97] adhere more to the text prompt, but they struggle with image similarity and image-text direction similarity. Our method, on the other hand, is able to achieve better image and image-text direction similarity.

4.2. User Study

We conduct an extensive user study using the Amazon Mechanical Turk (AMT) [6] platform, with the automatically generated test examples from Section 4.1. We compare all the baselines against our method using standard two-alternative forced-choice format. Users were given the input image, the edit text prompt, and two editing results (one from our method and one from the baseline). For each comparison, the users were asked to rate which editing result is better in terms of: (1) target prompt adherence, (2) input image preservation, (3) realism and (4) overall edit quality (*i.e.*, when taking all factors into account). As can be seen in Table 3, our method is preferred over the baselines in overall terms as well as the other terms. See the supplementary material for more details and statistical analysis.

4.3. Ablation Study

We conduct an ablation study for the following cases: (1) *Attention injection in all layers* — we perform the attention

Figure 8. **Qualitative Comparison.** We compare our method on real images against the baselines. SDEdit [52] faces challenges with preserving object identities and backgrounds (*e.g.*, rabbit and cat examples). P2P+NTI [34, 53] struggles with both preserving object identities (*e.g.*, rabbit and lion dolls examples) and adding new objects (*e.g.*, missing ball in the duck example and missing heart in the bottle example). Instruct-P2P [17] and MagicBrush [97] struggle with performing non-rigid editing (*e.g.*, raising of the paws in dog and cat examples, and the sitting of the rabbit in its example). MasaCTRL [18] has difficulty with preserving object identities (*e.g.*, cat, dog and lion doll examples) and adding new objects (*e.g.*, missing ball in the duck example and missing socks in the rabbit example). Our method, on the other hand, is able to adhere to the editing prompt while preserving the identities.

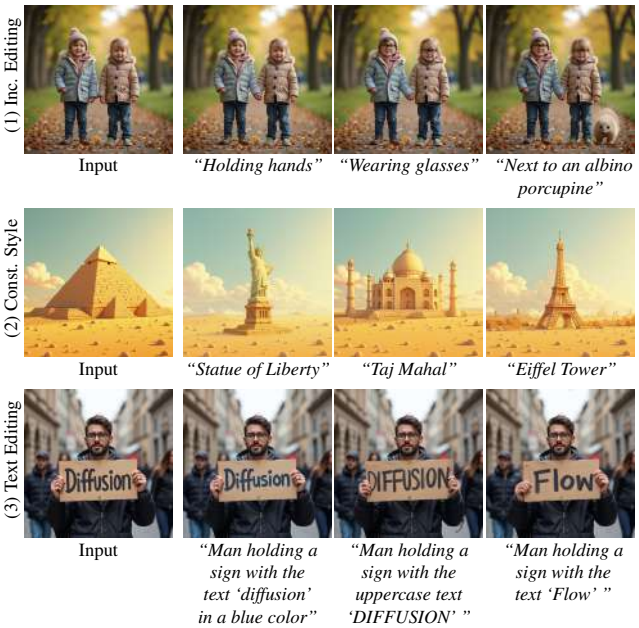


Figure 9. **Applications.** Our method can be used for various applications: (1) Incremental Editing — starting from a scene of two kids, the user can refine the image *iteratively* by making the kid hold hands, then wear glasses and finally add a porcupine next to them. (2) Consistent Style — starting from a scene with a given style, such as an animation of the Great Pyramid of Giza, the user can generate images of different places with the same style. (3) Text Editing — given a scene that contains text, our method is able to perform text-related editing such as color change, case change and text replacement.

injection that is described in Section 3.2 in all the layers (instead on the vital layers only). (2) *Attention injection non-vital layers* — we performed the attention injection in some non-vital layers (same amount of layers as vital layers). (3) *Attention extension* — instead of performing attention injection as described in Section 3.2, we extended [35, 82] the attention s.t. the generated images can attend to the reference image, as well as themselves. (4) *w/o latent nudging* — we omitted the latent nudging component (Section 3.3).

As can be seen in Table 2, we found that (1) performing attention injection in all the layers or performing (3) an attention extension in all the layers, significantly harms the text similarity. In addition, (2) performing an attention extension in the non-vital layers or (4) removing the latent nudging reduces the image similarity.

4.4. Applications

As demonstrated in Figure 9, our method can be used for various applications: (1) Incremental Editing — starting from a given scene, the user can refine the image *iteratively* in a step-by-step manner. (2) Consistent Style — starting from a scene with a given style, the user can generate other images in the same style [27, 35, 41]. (3) Text Editing — given a scene that contains text, our method is able to perform text-related editing such as color change, case change and text replacement.

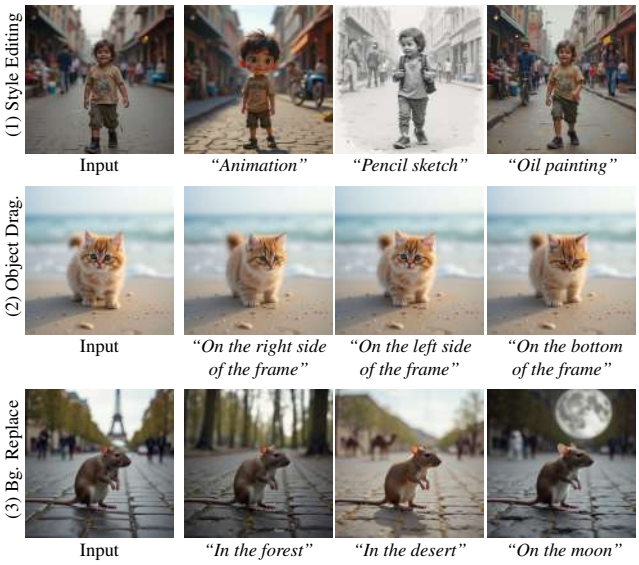


Figure 10. **Limitations.** Our method suffers from the following limitations: (1) Style Editing — given a photorealistic image of a boy, our method struggles with changing its style to an animation (the identity of the boy also changes), to pencil sketch (changes only to black&white) or to an oil painting (mainly makes the image smoother). (2) Object Dragging — given an image of a cat, our method is unable to drag it into different locations in the image, but changes the gaze of the cat instead. (3) Background Replacement — given an image of a rat on the road, our method unable to replace its background entirely (the road leaks).

5. Limitations and Conclusions

As demonstrated in Figure 10, our method suffers from the following limitations: (1) Style Editing — given an input image in one style (*e.g.*, photorealistic), our method struggles with changing it to a different style (*e.g.*, oil painting), as it relies on attention injection (Section 3.2). (2) Object Dragging — given an image with an object, our method is unable to drag it [11] into different locations in the image, as text-to-image models often struggle [10] with spatial prompt adherence. (3) Background Replacement — given an input image, our method struggles with replacing its background entirely with no leakage [22].

In conclusion, we present Stable Flow, a training-free method for image editing that enables *various* image editing tasks using the attention injection of the *same* vital layers group. We believe that our fully-automated approach of detecting vital layers may be also beneficial for other use-cases, such as generative models pruning and distillation. We hope that our layer analysis will inspire more work in the field of generative models and expanding the possibilities for creative expression.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020.

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Haim Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18490–18500, 2021. 3

[4] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *International Conference on Computer Graphics and Interactive Techniques*, 2023. 1

[5] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *ArXiv*, abs/2209.15571, 2022. 1

[6] Amazon. Amazon mechanical turk. https://www.mturk.com/, 2024. 6, 14

[7] Moab Arar, Andrey Voynov, Amir Hertz, Omri Avrahami, Shlomi Fruchter, Yael Pritch, Daniel Cohen-Or, and Ariel Shamir. Palp: Prompt aligned personalization of text-to-image models. 2024. 2

[8] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, 2022. 2

[9] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42(4), 2023. 2

[10] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18370–18380, 2023. 8

[11] Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. Diffuhaul: A training-free method for object dragging in images. *arXiv preprint arXiv:2406.01594*, 2024. 8, 21, 26

[12] Omri Avrahami, Amir Hertz, Yael Vinker, Moab Arar, Shlomi Fruchter, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. The chosen one: Consistent characters in text-to-image diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 2

[13] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *ArXiv*, abs/2204.02491, 2022. 2

[14] David Bau, Hendrik Strobelt, William S. Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 38:1 – 11, 2019. 3

[15] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *ArXiv*, abs/2103.10951, 2021. 2

[16] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolin'ario Passos. Ledits++: Limitless image editing using text-to-image models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8861–8870, 2023. 3

[17] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 6, 7, 13, 14, 15, 16

[18] 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 (ICCV)*, pages 22560–22570, 2023. 1, 2, 3, 4, 6, 7, 13, 14, 15, 16

[19] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 13, 15, 21, 23

[20] Hila Chefer, Shiran Zada, Roni Paiss, Ariel Ephrat, Omer Tov, Michael Rubinstein, Lior Wolf, Tali Dekel, Tomer Michaeli, and Inbar Mosseri. Still-moving: Customized video generation without customized video data. *ArXiv*, abs/2407.08674, 2024. 2

[21] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[22] Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be yourself: Bounded attention for multi-subject text-to-image generation. *ArXiv*, abs/2403.16990, 2024. 8

[23] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. *ArXiv*, abs/2408.00735, 2024. 3

[24] Tan M. Dinh, A. Tran, Rang Ho Man Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11379–11388, 2021. 3

[25] Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, abs/2403.03206, 2024. 1, 3, 13, 21, 36

[26] Johannes S. Fischer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan Andreas Baumann, Vincent Tao Hu, and Bjorn Ommer. Boosting latent diffusion with flow matching. *ArXiv*, abs/2312.07360, 2023. 1

[27] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. *ArXiv*, abs/2403.14572, 2024. 8

[28] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada. *ACM Transactions on Graphics (TOG)*, 41:1 – 13, 2021. 6, 14

[29] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. *ArXiv*, abs/2403.14602, 2024. 3

[30] 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

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[32] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Yuxiao Chen, Ding Liu, Qilong Zhangli, Anastasis Stathopoulos, Xiaoxiao He, Jindong Jiang, Zhaoyang Xia, Akash Srivastava, and Dimitris N. Metaxas. Proxedit: Improving tuning-free real image editing with proximal guidance. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4279–4289, 2023. 3

[33] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 3

[34] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 6, 7, 13, 14, 15, 16

[35] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4775–4785, 2023. 8

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, 2020. 1, 3

[37] Eliahu Horwitz, Jonathan Kahana, and Yedid Hoshen. Recovering the pre-fine-tuning weights of generative models. *ArXiv*, abs/2402.10208, 2024. 2

[38] Nisha Huang, Fan Tang, Weiming Dong, Tong-Yee Lee, and Changsheng Xu. Region-aware diffusion for zero-shot text-driven image editing. *ArXiv*, abs/2302.11797, 2023. 2

[39] 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*, abs/2402.17525, 2024. 2

[40] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv e-prints*, pages arXiv–2304, 2023. 3

[41] Jaeseok Jeong, Junho Kim, Yunjey Choi, Gayoung Lee, and Youngjung Uh. Visual style prompting with swapping self-attention. *ArXiv*, abs/2402.12974, 2024. 8

[42] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ArXiv*, abs/1603.08155, 2016. 15

[43] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 3

[44] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2

[45] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[46] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 2, 3, 6, 13, 21

[47] Shanglin Li, Bo-Wen Zeng, Yutang Feng, Sicheng Gao, Xuhui Liu, Jiaming Liu, Li Lin, Xu Tang, Yao Hu, Jianzhuang Liu, and Baochang Zhang. Zone: Zero-shot instruction-guided local editing. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6254–6263, 2023. 2

[48] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 6, 13, 15, 16, 17

[49] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022. 1

[50] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ArXiv*, abs/2209.03003, 2022. 1

[51] Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*, 2023. 3

[52] 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*, 2021. 2, 6, 7, 13, 14, 15, 16

[53] 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, 6, 7, 13, 14, 15, 16

[54] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021. 2

[55] Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michael Gharbi. Lazy diffusion transformer for interactive image editing. *ArXiv*, abs/2404.12382, 2024. 2

[56] OpenAI. ChatGPT. https://chat.openai.com/, 2022. Accessed: 2024-10-1. 3, 13, 21

[57] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023. 3, 13, 15, 23

[58] Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15866–15875, 2023. 3

[59] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multilayer selection for gan inversion and editing. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11389–11399, 2022. 3

[60] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[61] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2065–2074, 2021. 6, 14

[62] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22994–23004, 2023. 2

[63] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2022. 1, 2, 3

[64] Stanislav Pidhorskyi, Donald A. Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14092–14101, 2020. 3

[65] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T. Barron, Amit H. Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, C. Karen Liu, Lingjie Liu, Ben Mildenhall, Matthias Nießner, Bjorn Ommer, Christian Theobalt, Peter Wonka, and Gordon Wetzstein. State of the art on diffusion models for visual computing. *ArXiv*, abs/2310.07204, 2023. 2

[66] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. 2, 3

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 13, 15, 21, 23

[68] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[69] Omer Regev, Omri Avrahami, and Dani Lischinski. Click2mask: Local editing with dynamic mask generation. *ArXiv*, abs/2409.08272, 2024. 2

[70] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2287–2296, 2020. 3

[71] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42:1 – 13, 2021. 3

[72] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 3

[73] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. 1, 3

[74] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 3

[75] Mohammad Salama, Jonathan Kahana, Eliahu Horwitz, and Yedid Hoshen. Dataset size recovery from lora weights. *ArXiv*, abs/2406.19395, 2024. 2

[76] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *ArXiv*, abs/2311.10089, 2023. 2, 6

[77] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. 3

[78] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[79] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 2, 3

[80] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[81] Yoad Tewel, Rinon Gal, Dvir Samuel Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models, 2024. 2

[82] Yoad Tewel, Omri Kaduri, Rinon Gal, Yoni Kasten, Lior Wolf, Gal Chechik, and Yuval Atzmon. Training-free consistent text-to-image generation. *ArXiv*, abs/2402.03286, 2024. 8

[83] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40:1 – 14, 2021. 3

[84] 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. 2, 3

[85] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022. 2

[86] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022. 13, 21

[87] Bram Wallace, Akash Gokul, and Nikhil Vijay Naik. Edict: Exact diffusion inversion via coupled transformations. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22532–22541, 2022. 3

[88] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. *ArXiv*, abs/2305.18047, 2023. 2

[89] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18359–18369, 2023. 2

[90] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. 13

[91] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7589–7599, 2022. 1, 4

[92] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:3121–3138, 2021. 2

[93] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22428–22437, 2022. 2

[94] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18381–18391, 2022. 6

[95] Zhen Yang, Dinggang Gui, Wen Wang, Hao Chen, Bohan Zhuang, and Chunhua Shen. Object-aware inversion and reassembly for image editing. *ArXiv*, abs/2310.12149, 2023. 3

[96] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *ArXiv*, abs/1311.2901, 2013. 3

[97] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023. 2, 6, 7, 13, 14, 15, 16

[98] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 13, 15, 22, 23

[99] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 3

[100] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *ArXiv*, abs/2012.09036, 2020. 3

Stable Flow: Vital Layers for Training-Free Image Editing
Supplementary Material

Acknowledgments. We thank Omer Dahary for his valuable help and feedback. This work was supported in part by the Israel Science Foundation (grants 1574/21 and 2203/24).

A. Implementation Details

In Appendix A.1, we start by providing implementation details for our method. Next, in Appendix A.2, we provide the implementation details for the baselines we compared our method against. Later, in Appendix A.3, we provide the implementation details for the automatic evaluations dataset and metrics. Finally, in Appendix A.4 we provide the full details of the user study we conducted.

A.1. Method Implementation Details

As described in Section 3.1 of the main paper, we started by collecting a dataset of $K = 64$ text prompts using ChatGPT [56]. We instructed it to generate text prompts describing a diverse set of objects in different environments, with the focus on one main object. Then, we sampled K seeds denoted by S and used them to generate K corresponding images G_{ref}. Next, for each layer l, we bypass it by taking only the residual connection values. For each bypass, we generate K images using the same seed set S demoted by G_l. All the images were generated using Euler sampler in 15 steps and a guidance scale of 3.5.

Next, to evaluate the effect of each layer l on the final result, we compared the generated images G_l with their corresponding images G_{ref} using the DINOv2 [57] perceptual similarity metric. We term the layers that effect the generated image the most (*i.e.*, the layers with the lowest perceptual similarity) as vital layers, while the rest of the layers as non-vital layers. We found that the vital layers in the FLUX.1-dev model [46] are $[0, 1, 2, 17, 18, 25, 28, 53, 54, 56]$. For visualization results, please refer to Appendix B.5. We empirically found that layer 2 can be removed from this set. In addition, the vital layers for the Stable Diffusion 3 (SD3) [25] model vital layers are: $[0, 7, 8, 9]$. For more details, please refer to Appendix B.6.

A.2. Baselines Implementation Details

As explained in Section 4.1 of the main paper, we compare our method against the following baselines: SDEdit [52], P2P+NTI [34, 53], Instruct-P2P [17], MagicBrush [97], and MasaCTRL [18]. We reimplement SDEdit using the FLUX.1-dev model [46], and use the official implementation for the rest of the baselines.

We adapt the text prompts based on the baseline type: for SDEdit [52], P2P+NTI [34, 53], and MasaCTRL [18], we used the standard text prompt describing the desired edited scene (*e.g.*, *"A photo of a man with a red hat"*). For the instruction-based baselines Instruct-P2P [17] and MagicBrush [97] we adapted the style to fit an instructional format (*e.g.*, *"Make the person wear a red hat"*).

We used the following third-party implementations in this project:

- **FLUX.1-dev** model [46] HuggingFace Diffusers [86] implementation at `https://github.com/huggingface/diffusers`
- **P2P+NTI** [34, 53] official implementation at `https://github.com/google/prompt-to-prompt`
- **Instruct-P2P** [17] official implementation at `https://github.com/timothybrooks/instruct-pix2pix`
- **MagicBrush** [97] official implementation at `https://github.com/OSU-NLP-Group/MagicBrush`
- **MasaCTRL** [18] official implementation at `https://github.com/TencentARC/MasaCtrl`
- **DINOv2** [57] ViT-g/14 implementation by HuggingFace Transformers [90] at `https://github.com/huggingface/transformers`.
- **DINOv1** [19] ViT-B/16 implementation by HuggingFace Transformers [90] at `https://github.com/huggingface/transformers`.
- **CLIP** [67] ViT-L/14 implementation by HuggingFace Transformers [90] implementation at `https://github.com/huggingface/transformers`
- **LPIPS** [98] official implementation at `https://github.com/richzhang/PerceptualSimilarity`.

A.3. Automatic Metrics Implementation Details

As explained in Section 4.1 of the main paper, we prepare an evaluation dataset based on the COCO [48] validation dataset. We begin by filtering the dataset automatically to include at least one prominent non-rigid body. More specifically, we filter only images containing humans or animals that at least one of them is prominent enough, but not too small, *i.e.*, the prominent non-rigid body occupies at least 5% of the image but no more than 33%. Next, for each image, we apply various image editing tasks (non-rigid editing, object addition, object replacement, and scene editing) that take into account the prominent object from a list of different combinations, resulting in a total dataset of 3,200 samples. Examples of images from this dataset can be seen in Figure 13.

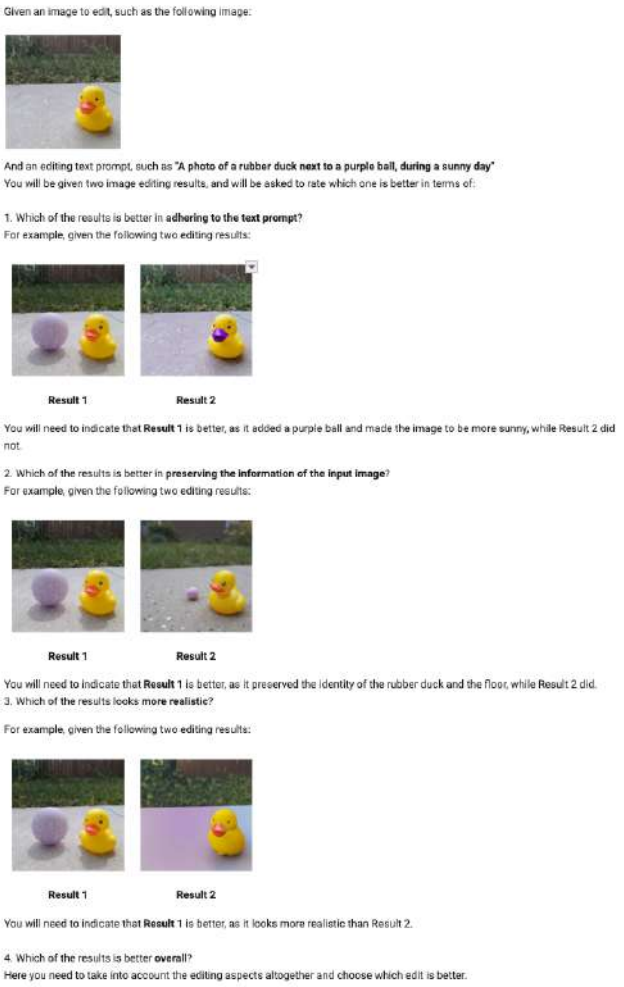


Figure 11. **User Study Instructions.** We provide the complete instructions for the user study we conducted using Amazon Mechanical Turk (AMT) [6] to compare our method with each baseline.

We evaluate the editing results using three metrics: (1) CLIP$_{img}$ which measures the similarity between the input image and the edited image by calculating the normalized cosine similarity of their CLIP image embeddings. (2) CLIP$_{txt}$ which measures the similarity between the edited image and the target editing prompt by calculating the normalized cosine similarity between the CLIP image embedding and the target text CLIP embedding. (3) CLIP$_{dir}$ [28, 61] which measures the similarity between the direction of the prompt change and the direction of the image change.

A.4. User Study Details

As described in Section 4.2 of the main paper, we conducted an extensive user study using the Amazon Mechanical Turk (AMT) [6] platform, using automatically generated test examples, as explained in Appendix A.3. We compared all the baselines with our method using a standard two-alternative forced-choice format. The users were given full instructions, as can be seen in Figure 11. Then, for each study trial, as shown in Figure 12, users were presented with an image and an instruction *"Given the following input image of a {CATEGORY}"* where {CATEGORY} is the COCO category of the prominent object. The users were given two editing results — one from our method and one from the baseline, and were asked the following questions:

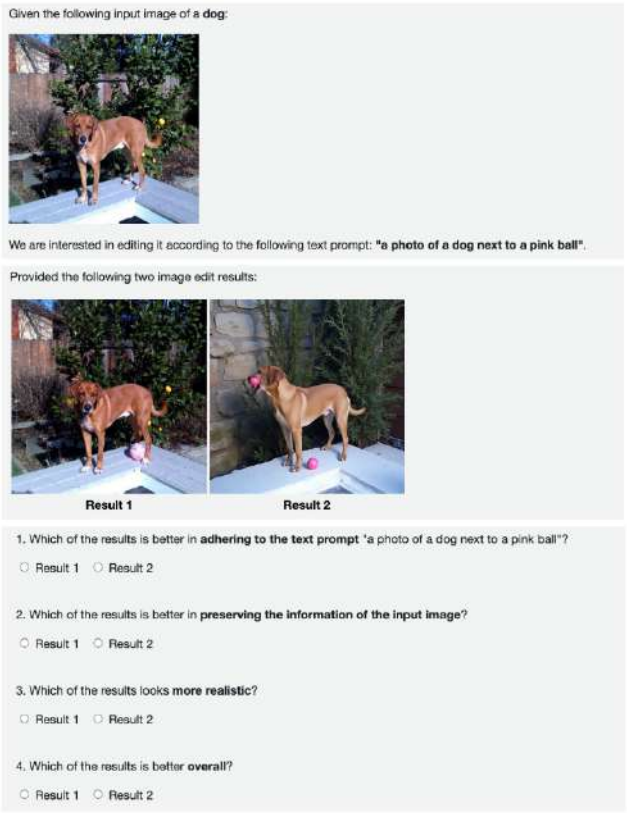


Figure 12. **User Study Trial.** We provide an example of a trial task in the user study conducted using Amazon Mechanical Turk (AMT) [6]. Users were asked four questions of a two-alternative forced-choice format. Complete instructions are shown in Figure 11.

Table 4. **User Study Statistical Significance.** A binomial statistical test of the user study results suggests that our results are statistically significant (p-value $< 5\%$).

Ours vs	Prompt Adher. p-value	Image Pres. p-value	Realism p-value	Overall p-value
SDEdit [52]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-6}$	$< 1e{-8}$
P2P+NTI [34, 53]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 6e{-8}$
Instruct-P2P [17]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 2e{-4}$
MagicBrush [97]	$< 5e{-5}$	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$
MasaCTRL [18]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$

1. *"Which of the results is better in adhering to the text prompt {PROMPT}?"*, where {PROMPT} is the editing target prompt.
2. *"Which of the results is better in preserving the information of the input image?"*
3. *"Which of the results looks more realistic?"*
4. *"Which of the results is better in overall?"*

We collected five ratings per sample, resulting in 320 ratings per baseline, for a total of 1,920 responses. The time allotted per task was one hour, to allow raters to properly evaluate the results without time pressure. A binomial statistical test of the user study results, as presented in Table 4, suggests that our results are statistically significant (p-value $< 5\%$).

B. Additional Experiments

In Appendix B.1, we start by providing additional comparisons and results of our method. Then, in Appendix B.2, we present experiments on using different perceptual metrics. Following that, in Appendix B.3, we test the effect of different sizes for vital layer set. Next, in Appendix B.4, we provide latent nudging experiments. Furthermore, in Appendix B.5 we present a full visualization of our layer bypassing method. Finally, in Appendix B.6, we test our method on the Stable Diffusion 3 backbone.

B.1. Additional Comparisons and Results

In Figure 13 we provide an additional qualitative comparison of our method against the baselines on real images extracted from the COCO [48] dataset, as explained in Section 4.1 in the main paper. As can be seen, SDEdit [52] struggles with preserving the object identities and backgrounds (*e.g.*, the bear and chicken examples). P2P+NTI [34, 53] struggles with preserving object identities (*e.g.*, the bear and person examples) and with adding new objects (*e.g.*, the missing hat in the sheep example and missing ball in the elephant example). Instruct-P2P [17] and MagicBrush [97] struggle with non-rigid editing (*e.g.*, the person raising hand example). MasaCTRL [18] struggles with preserving object identities (*e.g.*, the bear and person examples) and adding new objects (*e.g.*, the sheep and cat examples). Our method, on the other hand, is able to adhere to the editing prompt while preserving the identities.

Next, in Figure 14, we provide a qualitative comparison of the ablated cases that are explained in, Section 4.3 in the main paper. As can be seen, we found that (1) performing attention injection in all the layers or performing (3) an attention extension in all the layers, encourages the model to directly copy the input image while neglecting the target prompt. In addition, (2) performing an attention extension in the non-vital layers or (4) removing the latent nudging reduces the input image similarity significantly.

Finally, in Figures 15 and 16, we present additional image editing results using our method.

B.2. Different Perceptual Metrics

As explained in Section 3.1 of the main paper, we assess the impact of each layer by measuring the perceptual similarity between G_{ref} and G_ℓ using DINOv2 [57]. It raises the question of the importance of the specific perceptual [42] similarity metric when determining the vital layers.

To this end, we also experiment with different perceptual metrics: DINOv1 [19], CLIP [67], and LPIPS [98]. In Figures 18, 19 and 20 we plot the perceptual similarity per layer for each of these metrics. The vital layers, ordered by vitality, as defined in Equation 1 of the main paper, for each metric are:

- DINOv2 — $[1, 0, 2, 18, 53, 28, 54, 17, 56, 25]$.
- DINOv1 — $[1, 0, 2, 18, 53, 56, 54, 25, 28, 17]$.
- CLIP — $[2, 0, 1, 18, 53, 56, 54, 4, 17, 3]$.
- LPIPS — $[0, 1, 2, 18, 17, 56, 53, 54, 6, 4]$.

As can be seen, the vital set V is equivalent for DINOv2 and DINOv1 (even though there is a disagreement on the order). In addition, all the metrics include the set of $\{1, 0, 2, 18, 53, 54, 17, 56\}$ to be included in the vital set, while DINOv1 and DINOv2 suggest also including $\{28, 25\}$, CLIP suggests including $\{3, 4\}$ instead and LPIPS suggests including $\{6, 4\}$ instead. In Figure 21 we edited images with these slightly different vital layer sets, and found the differences to be negligible in practice.

B.3. Number of Vital Layers

The somewhat agnostic nature of our method to the specific perceptual metric, as described in Appendix B.2, raises the question of the importance of the entire vital layer set V to the editing task. To this end, in Figure 22 we experimented with omitting a growing number of vital layers and testing the editing results. As can be seen, when removing 80% of the vital layer set, the changes are negligible. However, when removing more than that, the editing results include unintended changes, such as identity changes (*e.g.*, man and woman examples) and background changes (*e.g.*, cat and blackboard examples). This is consistent with the results from Appendix B.2 that show that the least vital layers for each perceptual metric are less important for the image editing task.

B.4. Latent Nudging Experiments

As described in Section 3.3 of the main paper, we proposed using a latent nudging technique to avoid the bad reconstruction quality of vanilla inverse Euler ODE solver. We suggest multiplying the initial latent z_0 by a small scalar $\lambda = 1.15$ to slightly offset it from the training distribution. As shown in Figure 23, we empirically tested different values for the latent nudging hyperparameter λ. We performed

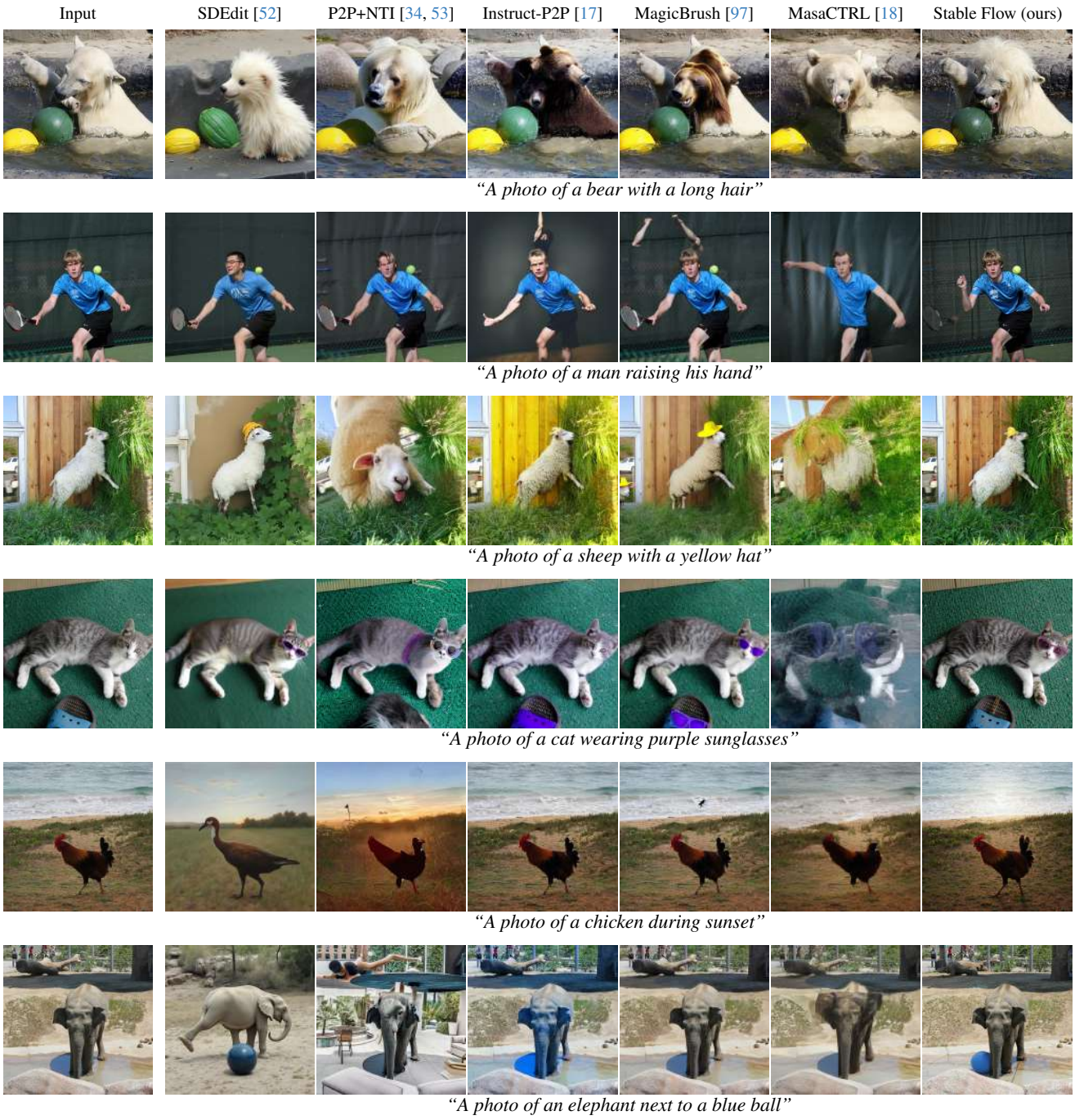


Figure 13. **Baselines Qualitative Comparison on Automatic Dataset.** As explained in Section 4.1 of the main paper, we compare our method against the baselines on real images extracted from the COCO [48] dataset. We find that SDEdit [52] struggles with preserving the object identities and backgrounds (*e.g.*, bear and chicken examples). P2P+NTI [34, 53] struggles with preserving object identities (*e.g.*, bear and person examples) and with adding new objects (*e.g.*, missing hat in the sheep example and missing ball in the elephant example). Instruct-P2P [17] and MagicBrush [97] struggle with non-rigid editing (*e.g.*, person raising hand). MasaCTRL [18] struggles with preserving object identities (*e.g.*, bear and person examples) and adding new objects (*e.g.*, sheep and cat examples). Our method, on the other hand, is able to adhere to the editing prompt while preserving the identities.

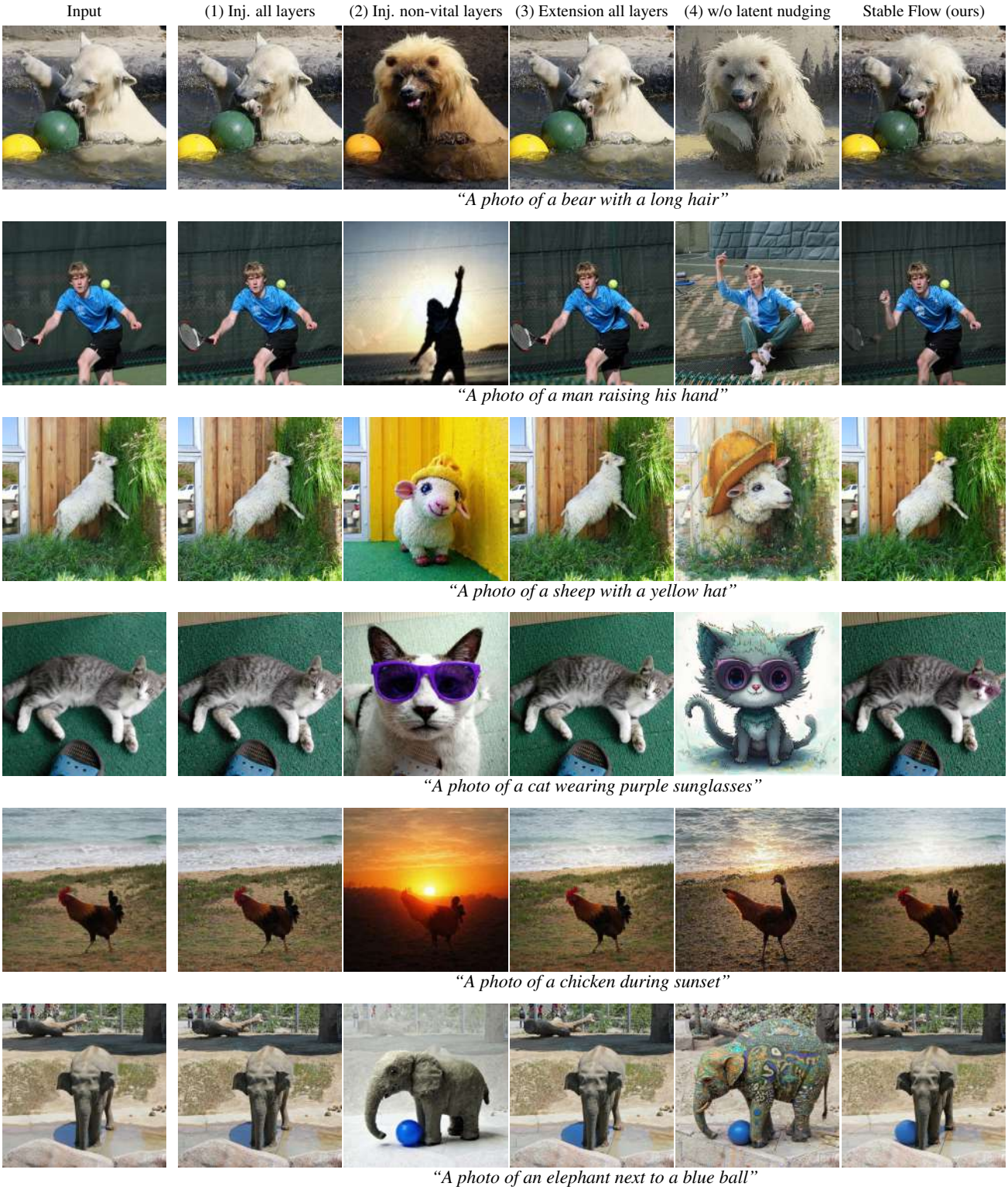


Figure 14. **Ablations Qualitative Comparison on Automatic Dataset.** As explained in Section 4.3 of the main paper, we compare our method against several ablation cases on real images extracted from the COCO [48] dataset. As can be seen, we found that (1) performing attention injection in all the layers or performing (3) an attention extension in all the layers encourages the model to directly copy the input image while neglecting the target prompt. In addition, (2) performing an attention extension in the non-vital layers or (4) removing the latent nudging reduces the input image similarity significantly.

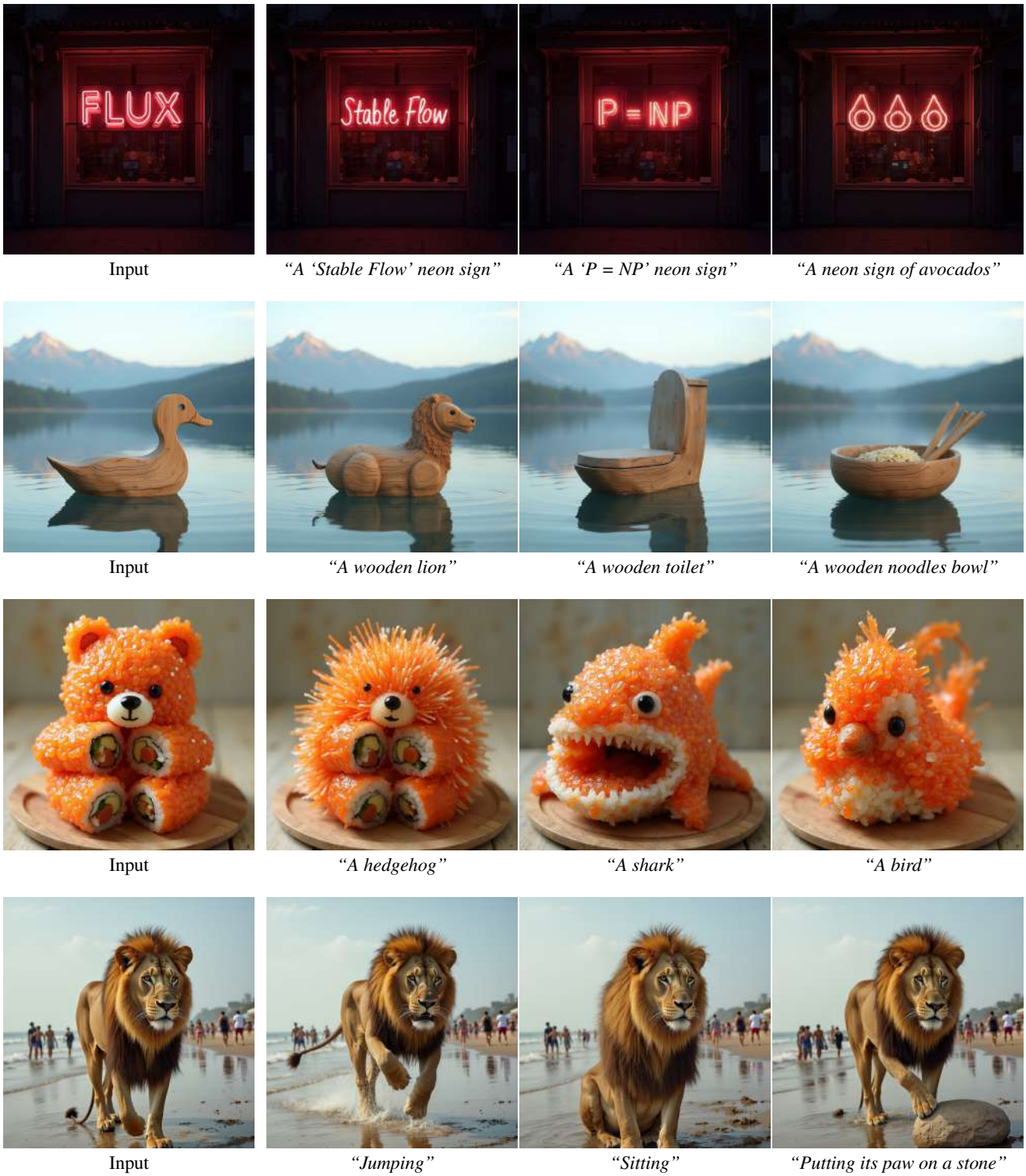


Figure 15. **Additional Results.** We provide various editing results of our method. These different edits are done using the *same* vital layer set.

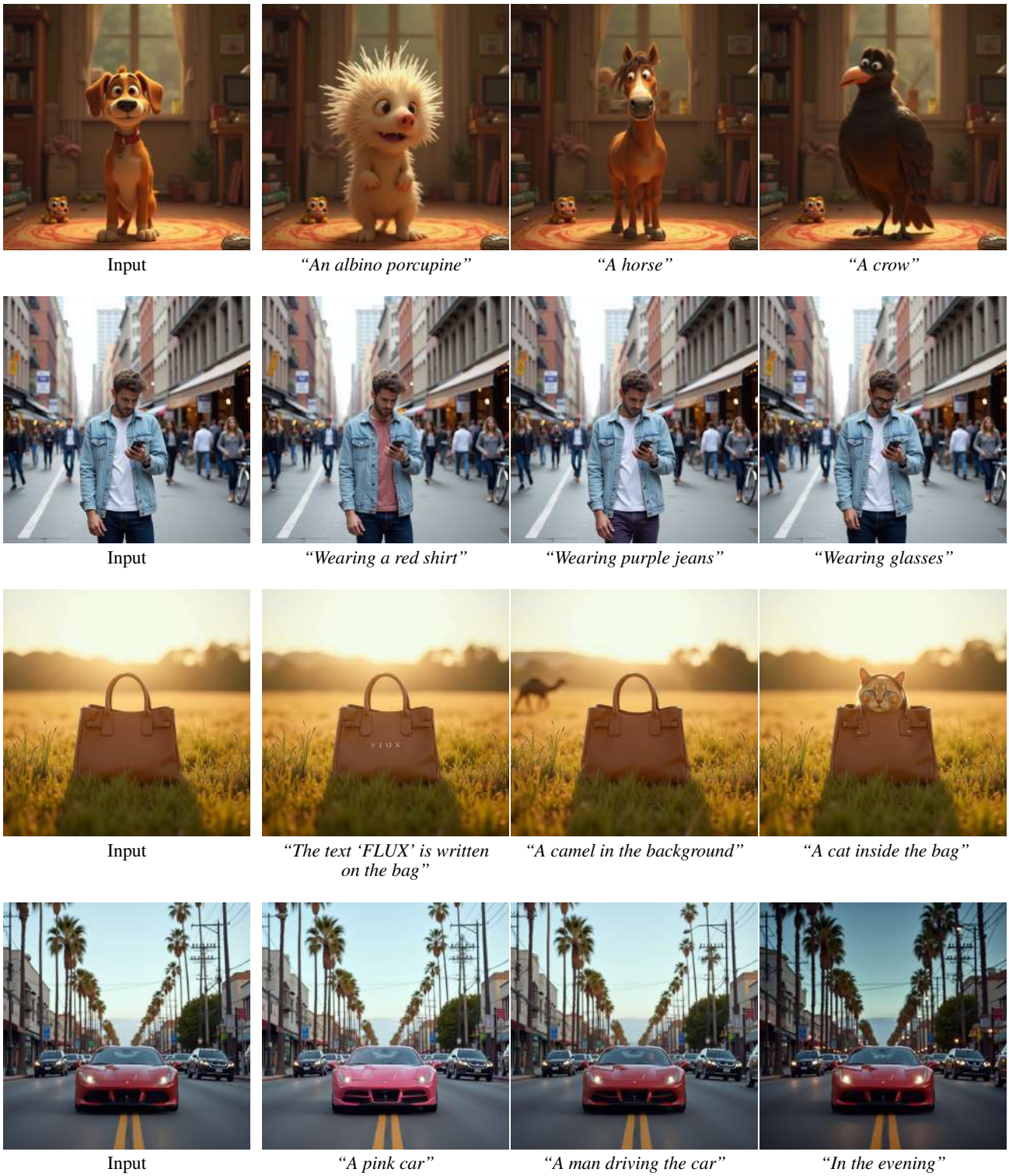
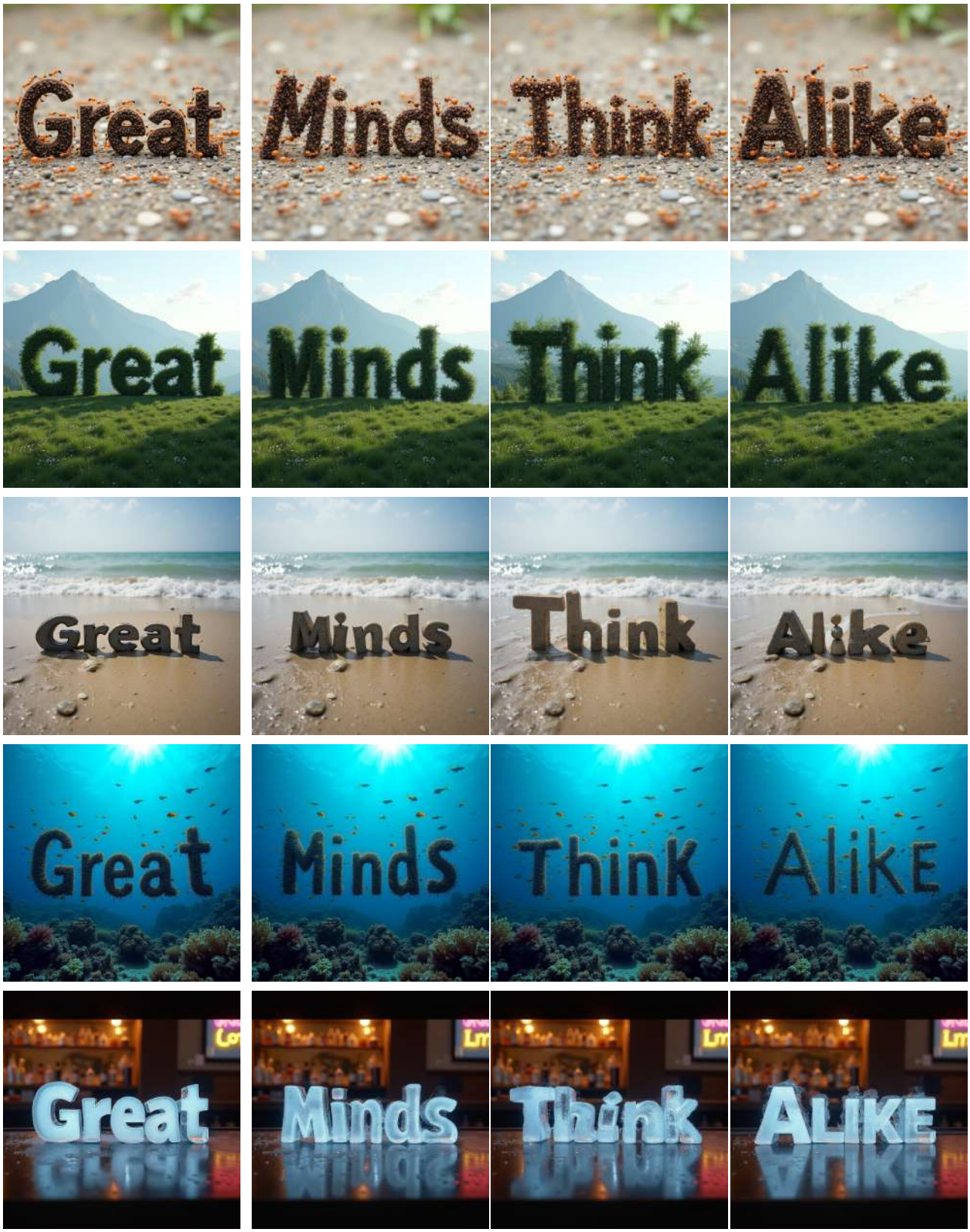


Figure 16. **Additional Results.** We provide various editing results of our method. These different edits are done using the *same* vital layer set.



Input *"Minds"* *"Think"* *"Alike"*

Figure 17. **Additional Results.** Given an input image that contain a text, our method cat edit the text while keeping the background and style.

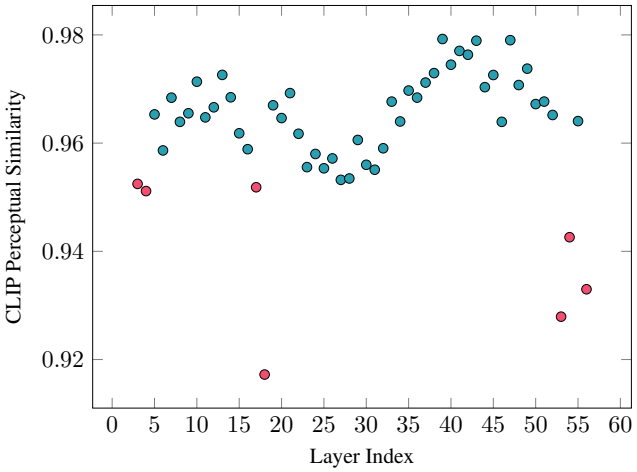


Figure 18. **Layer Removal Quantitative Comparison Using CLIP.** As explained in Appendix B.2, we measured the effect of removing each layer of the model by calculating the *CLIP* [67] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

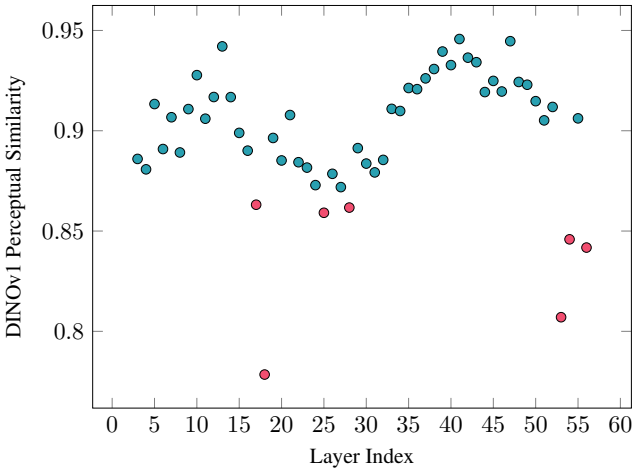


Figure 19. **Layer Removal Quantitative Comparison Using DINOv1.** As explained in Appendix B.2, we measured the effect of removing each layer of the model by calculating the *DINOv1* [19] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

inversion using the inverse Euler ODE solver with a high number of 1,000 inversion (and denoising) steps, to reduce the inversion error. However, even when using such a high number of inversion/denoising steps, we notice that when not using latent nudging (*i.e.*, $\lambda = 1.0$), the reconstruction quality is poor (notice the eyes and the legs of the dog). Next, we found that $\lambda = 1.15$ is the smallest value that enables full reconstruction using the inverse Euler solver. Furthermore, nudging values that are too high (*e.g.*, $\lambda = 3.0$) result in saturated images. Lastly, we notice that decreasing nudging values (*i.e.*, $\lambda < 1.0$) severely damages the reconstruction quality.

In addition, we experiment with a simpler inversion variant based on latent caching (termed *DDPM bucketing* in DiffUHaul [11]), in which we saved the series of latents during the inversion process without applying latent nudging. As shown in Figure 24, this approach indeed achieves perfect inversion (second column), but (third column) still struggles with preserving the identities while editing the image (*e.g.*, the rabbit and duck examples) or significantly alters the image (*e.g.*, the cat and man examples). On the other hand, our method (fourth column) with the latent nudging is able to preserve the identities during editing. In practice, we found that using latent caching in addition to latent nudging enables inversion with a lower number of steps (50 steps), hence, this is the approach we used.

B.5. Layer Bypassing Visualization

As explained in Section 3.1 of the main paper, to quantify layer importance in FLUX model, we devised a systematic evaluation approach. Using ChatGPT [56], we automatically generated a set P of $k = 64$ diverse text prompts, and draw a set S of random seeds. Each of these prompts was used to generate a reference image, yielding a set G_{ref}. For each DiT layer $\ell \in \mathbb{L}$, we performed an ablation by bypassing the layer using its residual connection. This process generated a set of images G_ℓ from the same prompts and seeds.

In Figures 25–32, we provide a full visualization of the reference set G_{ref} along with the generation sets $G_0 - G_{56}$. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions.

B.6. Stable Diffusion 3 Results

All the experiments in the main paper were based on the FLUX.1-dev [46] model. We also experimented with a different DiT text-to-image flow model named Stable Diffusion 3 [25] based on the Diffusers [86] implementation of the medium model.

As described in Section 3.1 of the main paper, we mea-

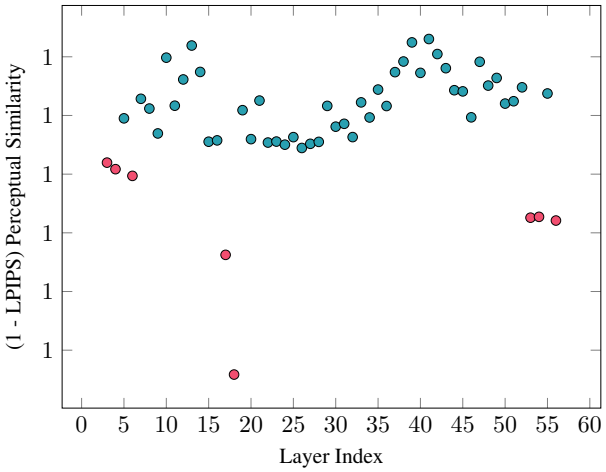


Figure 20. **Layer Removal Quantitative Comparison Using LPIPS.** As explained in Appendix B.2, we measured the effect of removing each layer of the model by calculating the *(1 - LPIPS)* [98] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

sured the importance of each of the layers of this model. As shown in Figure 33, we measured the effect of removing each layer from the model by calculating the perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact.

Next, in Figure 34 we illustrate the qualitative differences between vital and non-vital layers. While bypassing non-vital layers (G_1 and G_{21}) results in modest alterations, bypassing vital layers leads to significant changes: complete noise generation (G_0) or severe distortions (G_7, G_8, and G_9).

Finally, in Figure 35, we perform various editing operations using the same mechanism of injecting the reference image information into the vital layers of the model, as described in Section 3.2 of the main paper.

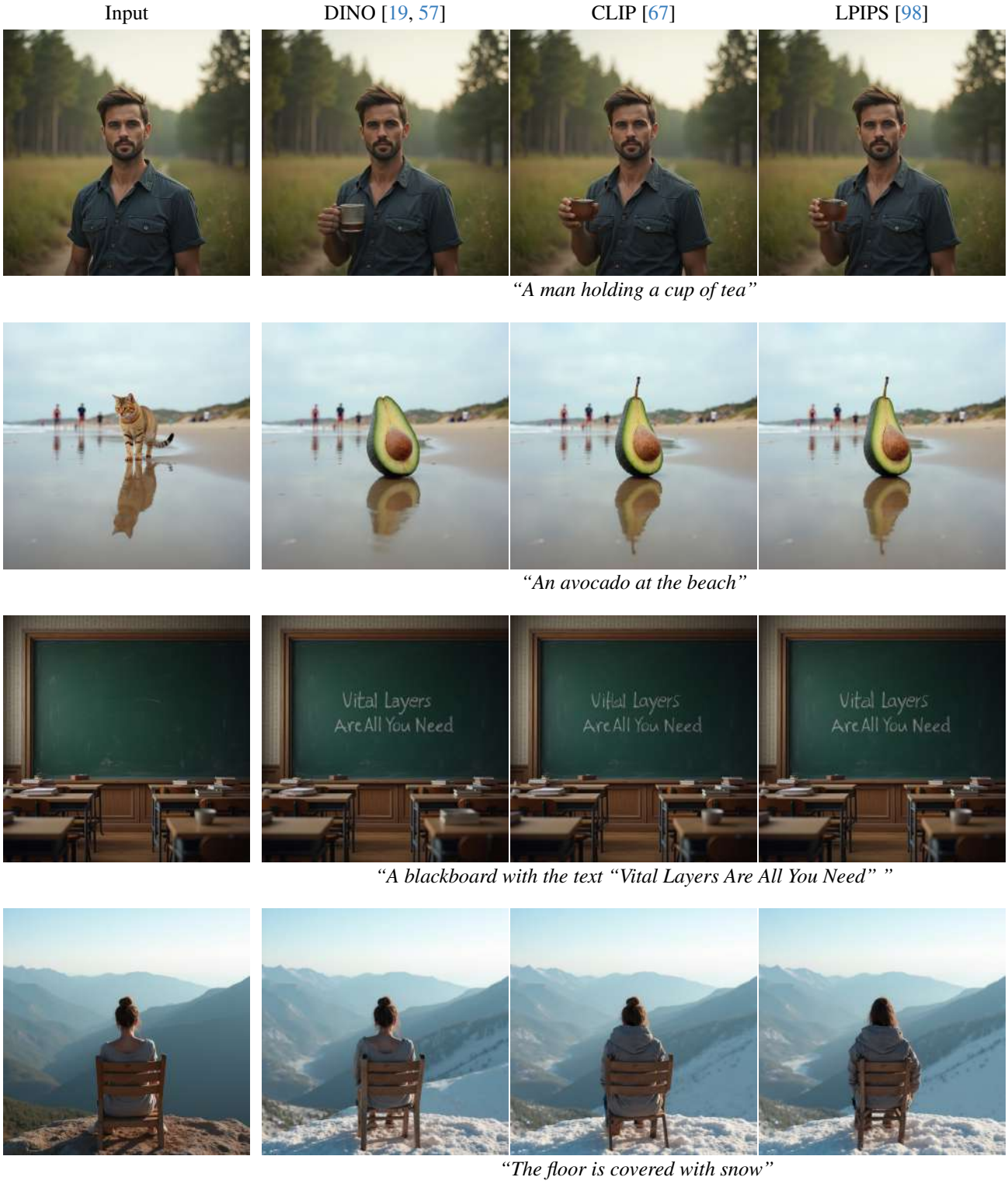


Figure 21. **Metrics Qualitative Comparison.** As described in Appendix B.2, we also experimented with other perceptual metrics. We found DINOv2 [57] and DINOv1 [19] to produce the same set of vital layer. While CLIP [67] and LPIPS [98] replaced two layers in the vital layers set (though they include most of the vital layer set as in DINO). As can be seen, the differences between these sets are negligible when editing images.

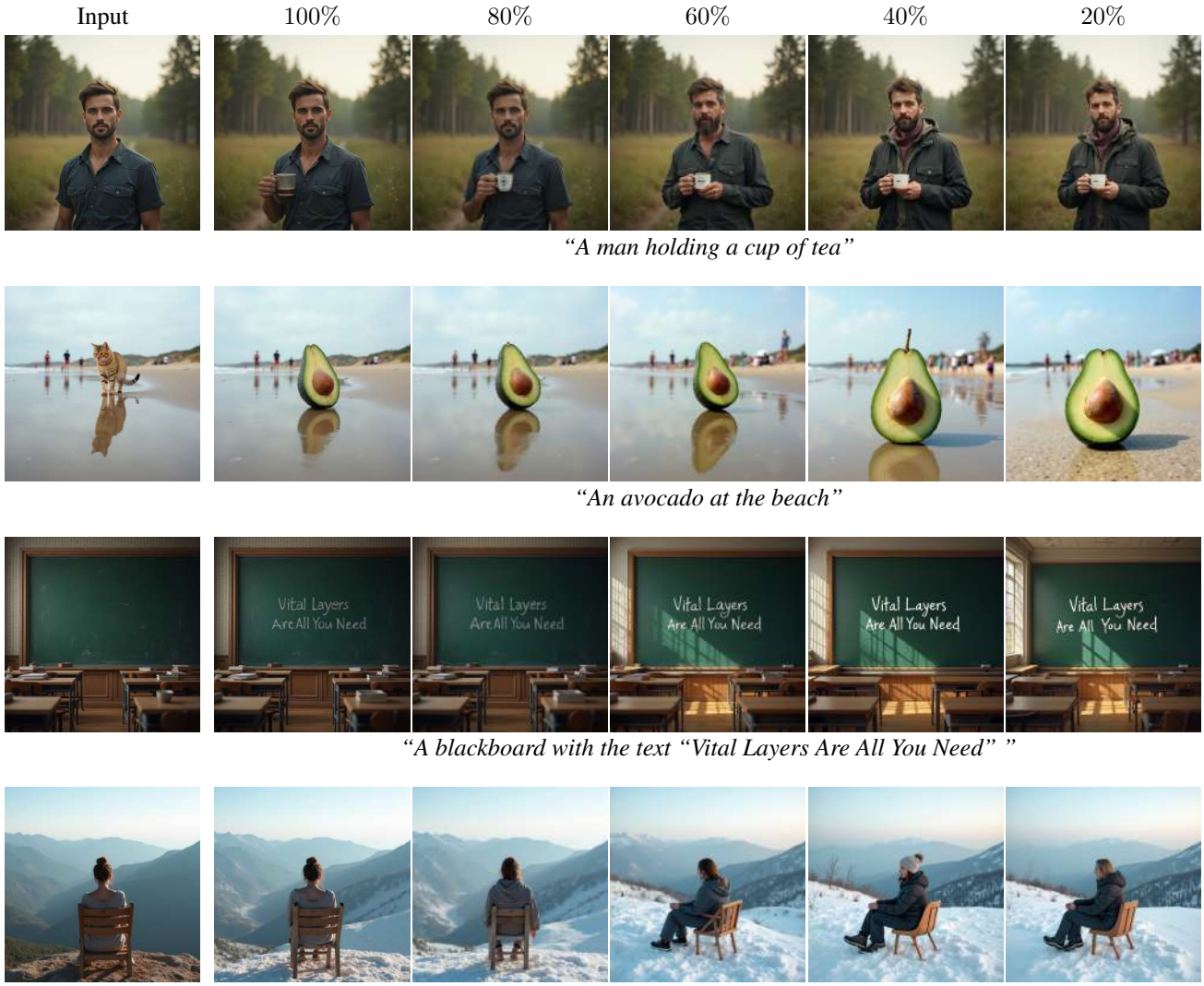


Figure 22. **Number of Vital Layers Comparison.** As explained in Appendix B.3, we experimented with choosing a different portion of the calculated vital layer set V. As can be seen, when removing 80% of the vital layer set, the changes are negligible. However, when removing more than that, the editing results include unintended changes, such as identity changes (*e.g.*, the man and woman examples) and background changes (*e.g.*, the cat and blackboard examples).

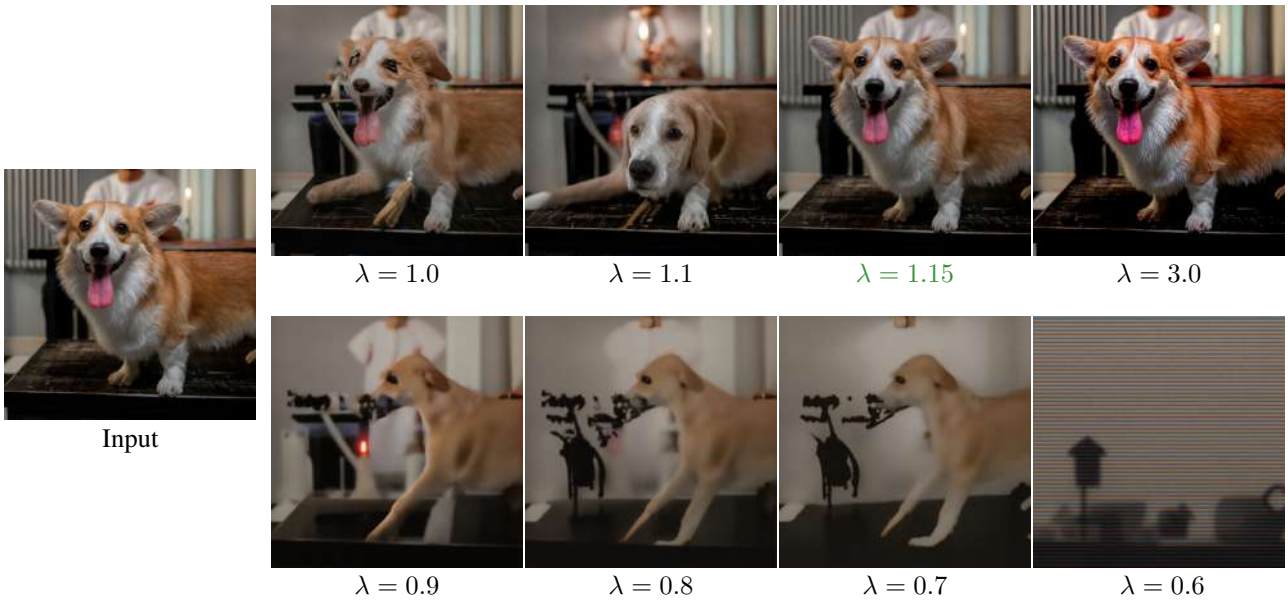


Figure 23. **Latent Nudging Values.** As described in Appendix B.4, we empirically tested different values for the latent nudging hyper-parameter λ. In our experiments, we performed inversion using the inverse Euler ODE solver with a high number of 1,000 inversion (and denoising) steps, to reduce the inversion error. However, even when using such a high number of inversion/denoising steps, we notice that when not using latent nudging (*i.e.*, $\lambda = 1.0$), the reconstruction quality is poor (notice the eyes and the legs of the dog). Next, we found that $\lambda = 1.15$ is the smallest value that enables full reconstruction using the inverse Euler solver. Furthermore, nudging values that are too high (*e.g.*, $\lambda = 3.0$) result in saturated images. Lastly, we notice that reducing nudging values ($\lambda < 1.0$) severely damages the reconstruction quality.

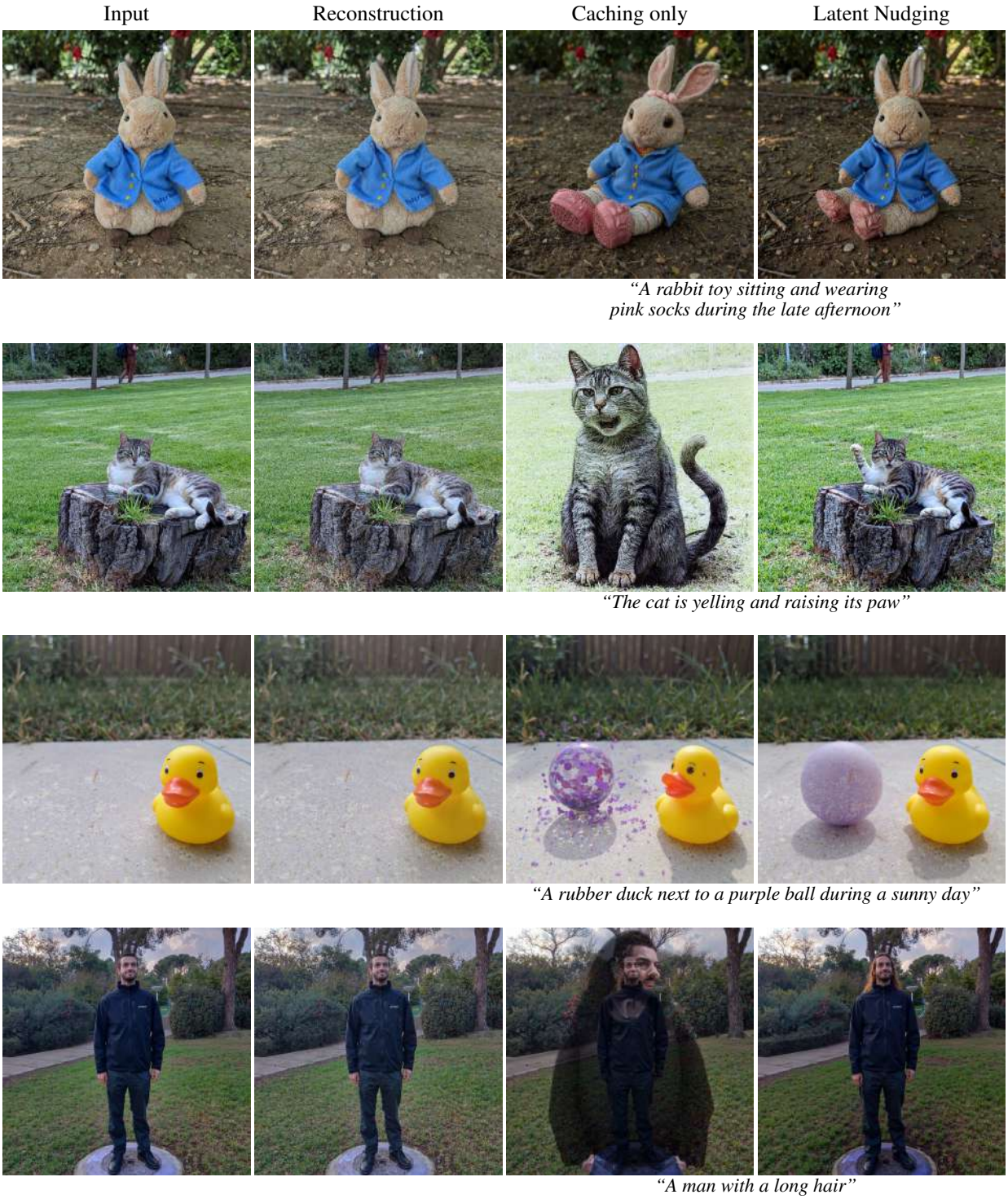


Figure 24. **Latent Caching.** As explained in Appendix B.4, we also tested a latent caching approach [11], in which we saved the series of latents during the inversion process without applying latent nudging. As can be seen, this approach indeed achieves perfect inversion (second column), but (third column) still struggles with preserving the identities while editing the image (*e.g.*, the rabbit and duck examples) or significantly alters the image (*e.g.*, the cat and man examples). On the other hand, our method with the latent nudging (fourth column) is able to preserve the identities during editing.

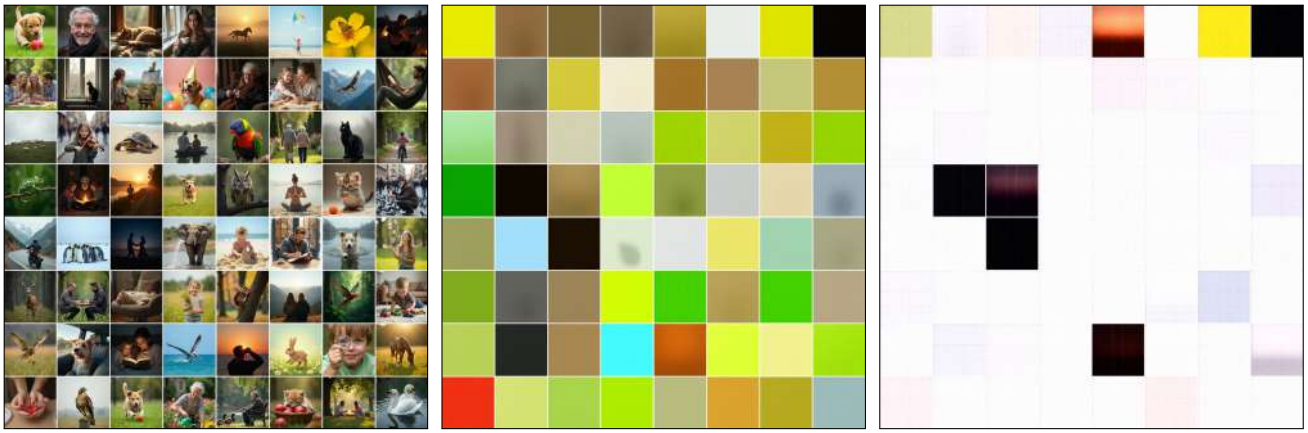


Figure 25. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_0 – G_2 are vital layers, while G_3 – G_7 are non-vital layers.

Figure 26. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_8 – G_{15} are all non-vital layers.

G_{ref} G_{16} G_{17}

G_{18} G_{19} G_{20}

G_{21} G_{22} G_{23}

Figure 27. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{17} and G_{18} are vital layers, while G_{16} and G_{19} – G_{23} are non-vital layers.

G_{ref} G_{24} G_{25}

G_{26} G_{27} G_{28}

G_{29} G_{30} G_{31}

Figure 28. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{25} and G_{28} are vital layers, while G_{24}, G_{26} – G_{27} and G_{29} – G_{31} are non-vital layers.

G_{ref} G_{32} G_{33}

G_{34} G_{35} G_{36}

G_{37} G_{38} G_{39}

Figure 29. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{31} – G_{39} are non-vital layers.

G_{ref} G_{40} G_{41}

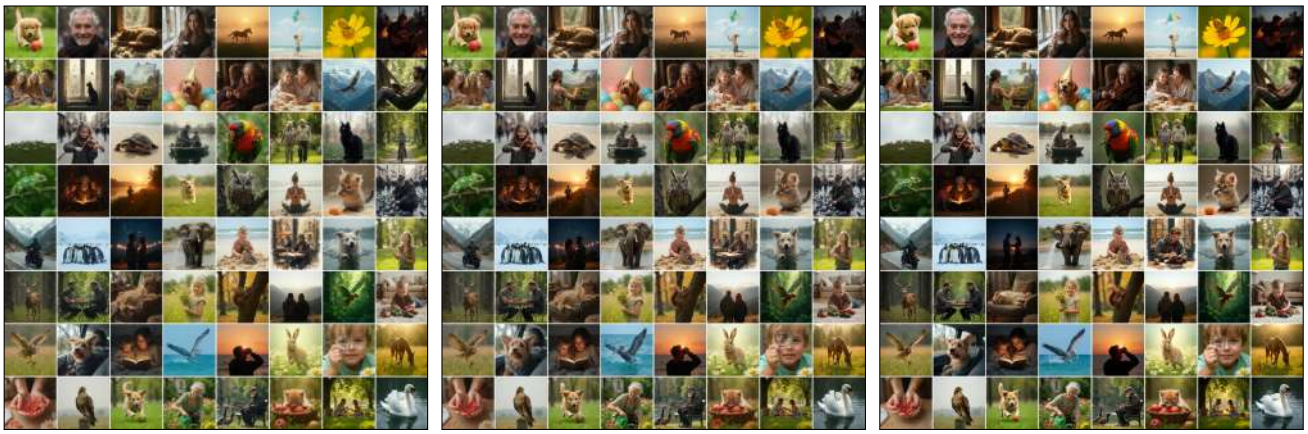
G_{42} G_{43} G_{44}

G_{45} G_{46} G_{47}

Figure 30. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{40} – G_{47} are non-vital layers.

G_{ref} G_{48} G_{49}

G_{50} G_{51} G_{52}



G_{53} G_{54} G_{55}

Figure 31. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{53} – G_{54} are vital layers, while G_{48} – G_{52} and G_{55} are non-vital layers.

G_{ref} G_{56}

Figure 32. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Appendix B.5. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{56} is a vital layer.

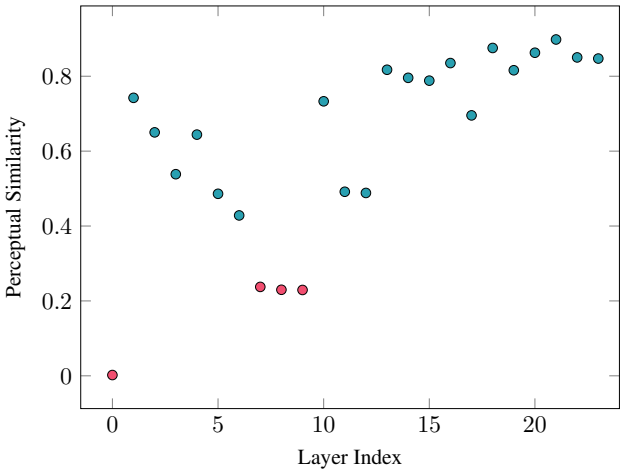


Figure 33. **Layer Removal Quantitative Comparison Stable Diffusion 3.** As explained in Appendix B.6, we measured the effect of removing each layer of the model by calculating the perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. For a visual comparison, please refer to Figure 34.

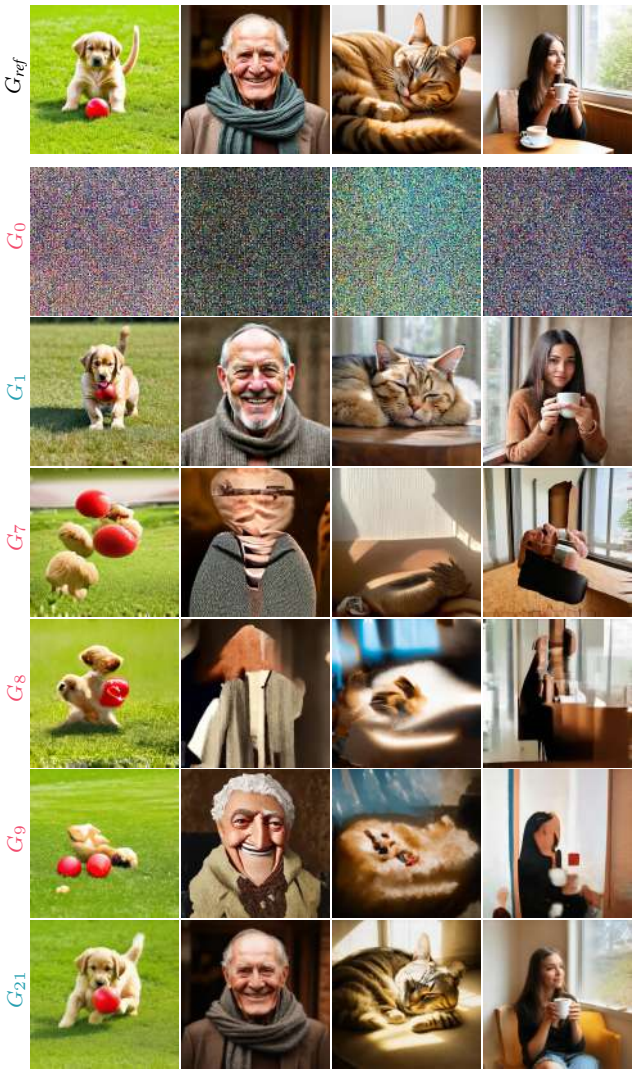


Figure 34. **Layer Removal Qualitative Comparison Stable Diffusion 3.** As explained in Appendix B.6, we illustrate the qualitative differences between vital and non-vital layers. While bypassing non-vital layers (G_1 and G_{21}) results in modest alterations, bypassing vital layers leads to significant changes: complete noise generation (G_0), or severe distortions (G_7, G_8 and G_9). For a quantitative comparison, please refer to Figure 33

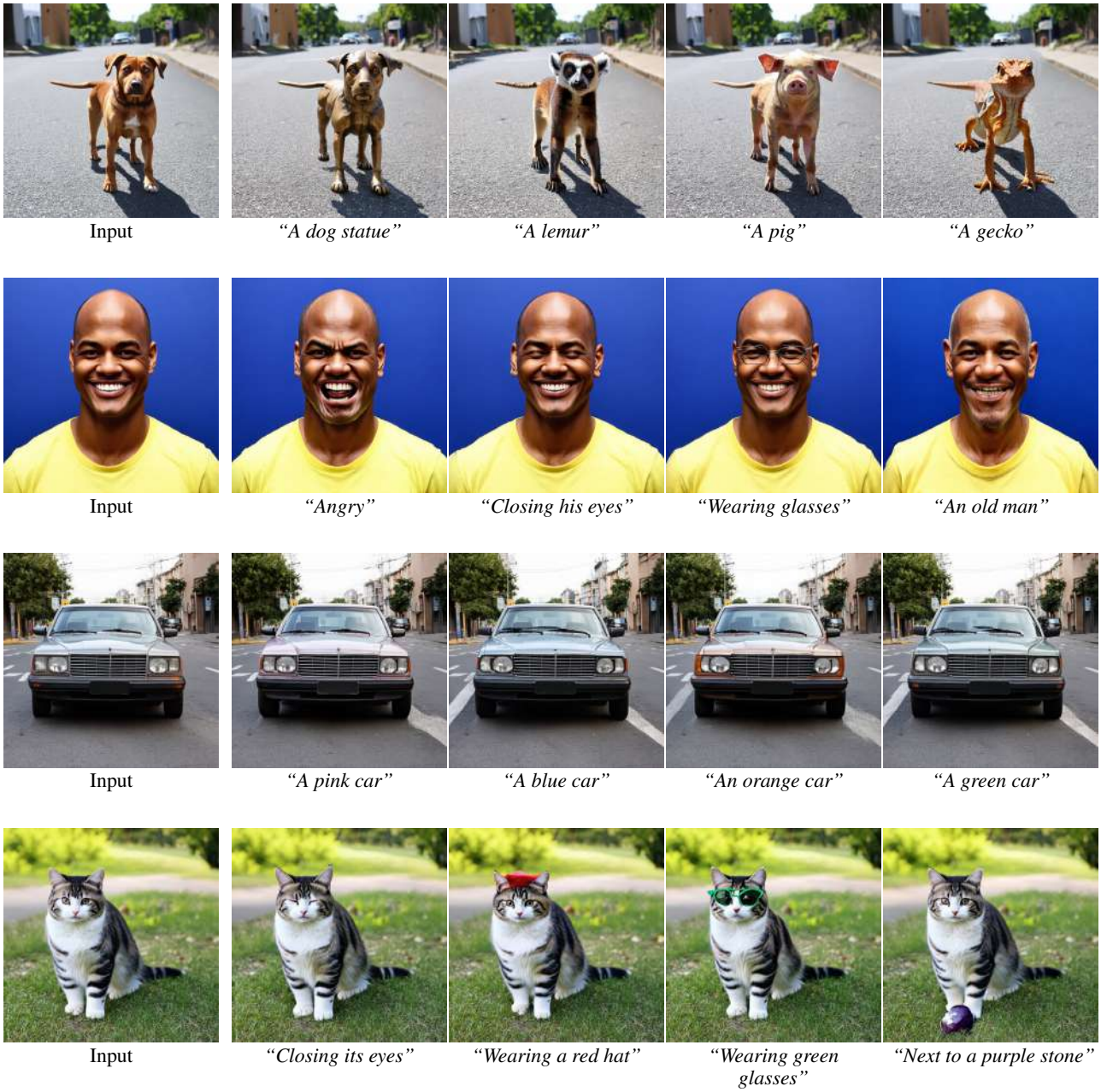


Figure 35. **Stable Diffusion 3 Editing Results.** As explained in Appendix B.6, we tested our Stable Flow method on the Stable Diffusion 3 backbone [25]. As can be seen, we are able to perform various editing operations using the same mechanism of injecting the reference image information into the vital layers of the model.