

SUPPLEMENTARY MATERIAL

A ADDITIONAL RESULTS

Additional results obtained by our method for image editing with FLUX are presented in Fig. S1.

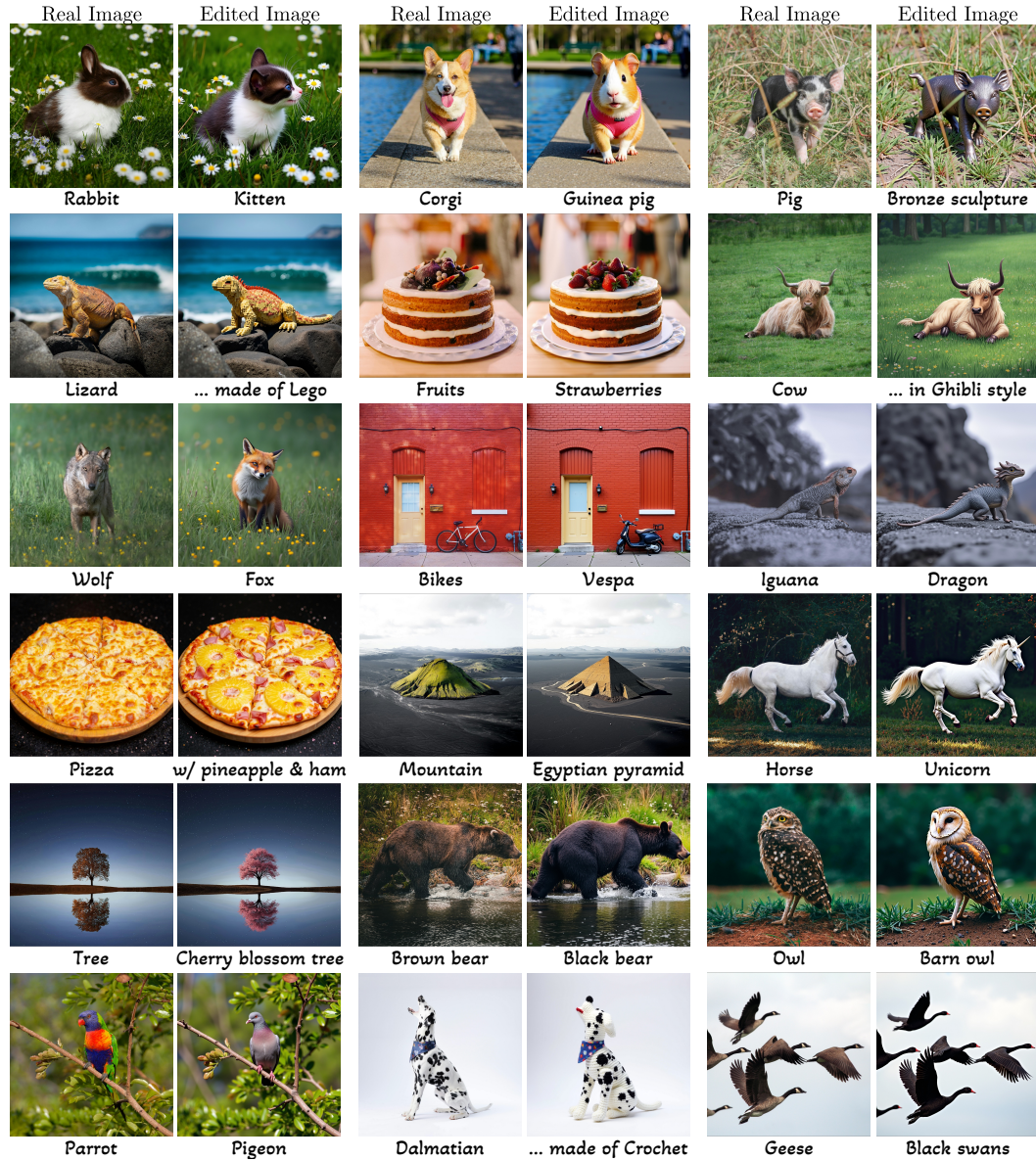


Figure S1: Additional FlowOpt results (FLUX).

B COMPARISONS

B.1 IMAGE RECONSTRUCTION (INVERSION)

Figure S2 displays the results for the unconditional case for FLUX, evaluated by pixel-space RMSE, PSNR, SSIM and LPIPS as a function of the NFEs. The left column in this figure is obtained by using CFG = 1, and the right column is obtained by using CFG = 0. The details of this experiments are the same as detailed in Sec. 5.

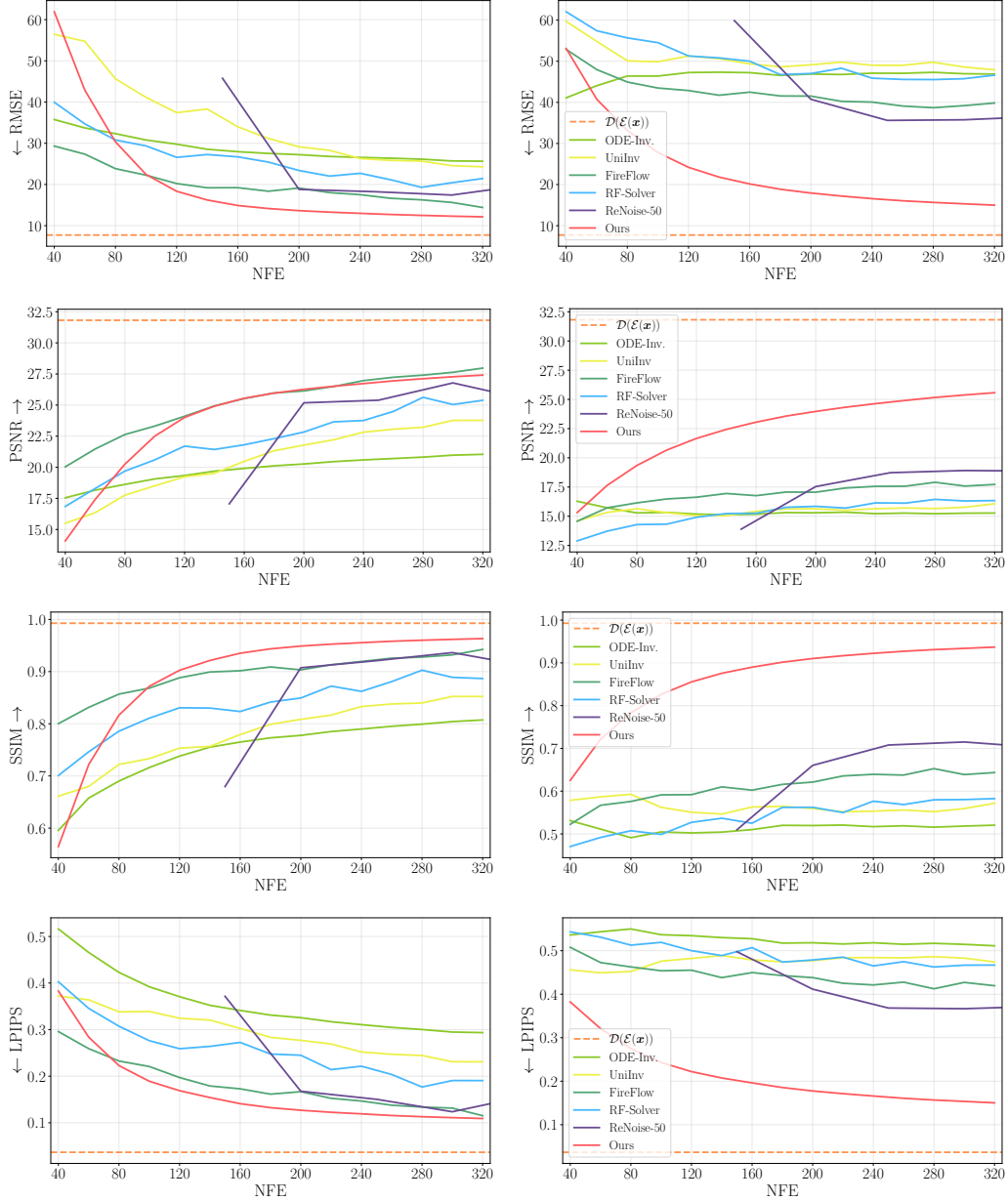


Figure S2: Reconstruction quantitative comparisons (FLUX). Pixel-space RMSE (first row), PSNR (second row), SSIM (third row), and LPIPS (last row) as a function of the number of NFEs for several inversion methods, for unconditional sampling with CFG = 1 (left) and with CFG = 0 (right). The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model.

B.2 IMAGE EDITING

B.2.1 ADDITIONAL QUALITATIVE COMPARISONS

Figure S3 presents additional comparisons on image editing. We can see that FlowOpt achieves consistently the best results both in terms of source image adherence and in terms of text adherence. For example, in the third row our method is the only one that managed to preserve the background and the structure of the rocks in the foreground. Similarly, in the fifth row, our method is the only one that preserved the posture of the dogs, and in the last row when replacing the sitting man with a golden sculpture of Buddha, FlowOpt preserves the original limb orientations and scene background.

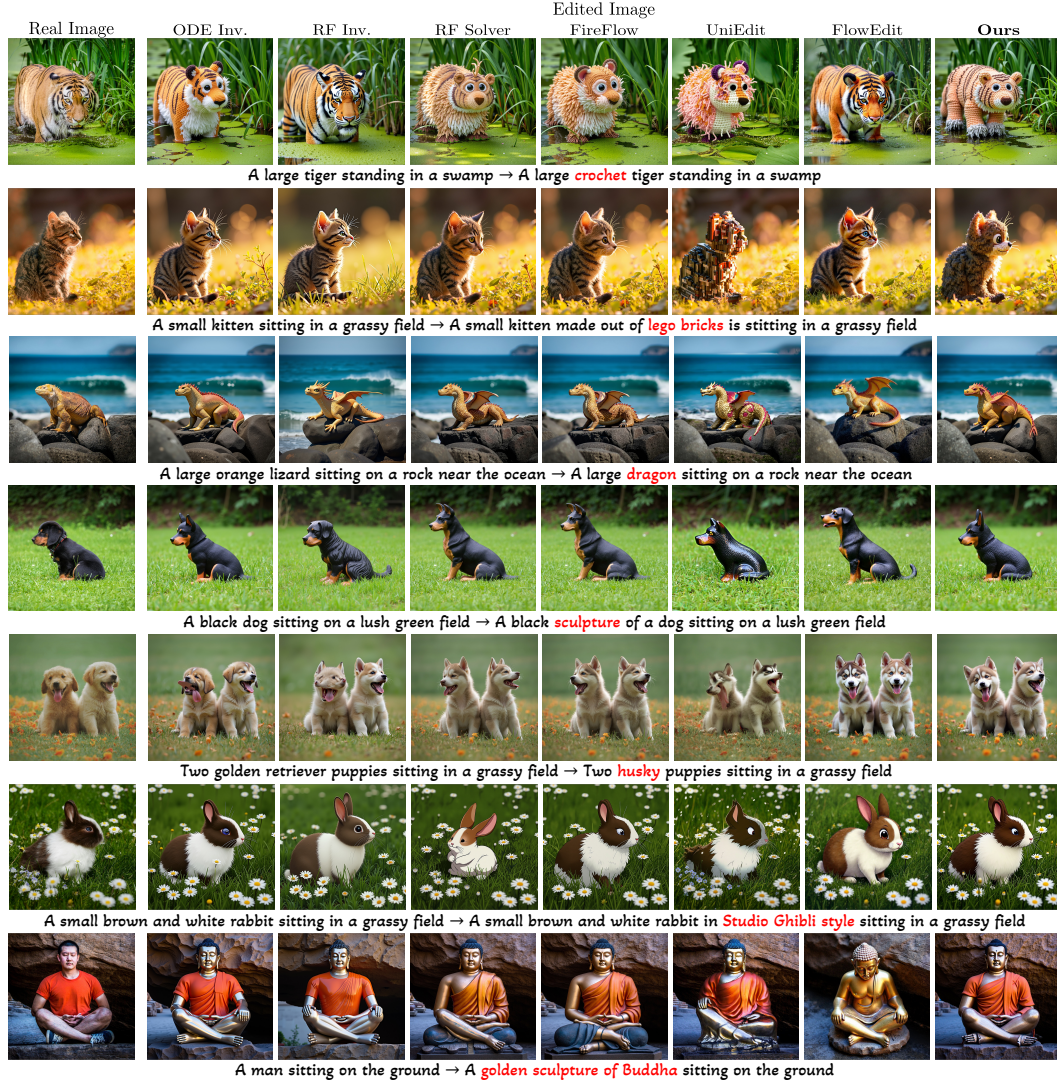


Figure S3: Additional qualitative comparisons (FLUX). Fine details are visible upon zooming in.

B.2.2 DETAILS OF THE EXPERIMENT SETTINGS

Figure 10 compares between all methods in terms of text adherence (CLIP Text) and image adherence measures (CLIP Image, DINOv3 and DreamSim). Figure S4 provides a more detailed comparison between all methods.

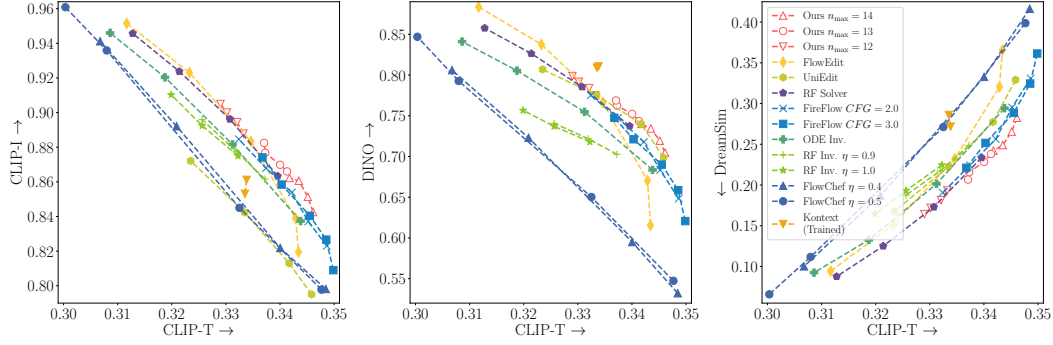


Figure S4: Editing quantitative comparisons (FLUX). Text adherence is measured by CLIP-Text (x-axis) for all figures. Image adherence (y-axis) is measured by CLIP-Image (left), DINOv3 (center), and DreamSim (right). Connected markers represent different hyperparameters.

FLUX hyperparameters. Table S1 lists the settings with which FlowEdit, ODE Inversion, and FlowOpt were run in Fig. S4. The hyperparameters for all figures in the main text (except for Fig. 1) are marked in bold in this table.

Table S1: FLUX hyperparameters.

	T	n_{\max}	CFG @ source	CFG @ target	N iterations
FlowEdit	28	27, 26, 24 , 22, 20	1.5	5.5	-
ODE Inversion	50	45, 40 , 35, 30	1	3.5	-
FlowOpt	15	14, 13 , 12	1	3.5	2, 3, 4, 5

For UniEdit, the evaluated hyperparameters are presented in Tab. S2, with the chosen value for their α parameter marked in bold. In our notation, $\alpha = n_{\max}/T$.

Table S2: FLUX UniEdit hyperparameters.

T	α delay rate	ω guidance scale
15	$\frac{2}{5}$, $\frac{2}{3}$, $\frac{11}{15}$, $\frac{3}{5}$	5

For RF-Solver and FireFlow, the hyperparameters that were evaluated are presented in Tab. S3, following their official implementation.

Table S3: RF-Solver and FireFlow hyperparameters.

	T	CFG	Injection step
RF-Solver	15	2	2 , 3, 4, 5
FireFlow	30	2 , 3	1, 2, 3, 4 , 5

For RF-Inversion, the paper provides a set of hyperparameters for each kind of editing. We evaluate the sets of hyperparameters provided in their supplementary material. These are reported in Tab. S4. We choose the set that achieved the best results.

Table S4: RF-Inversion hyperparameters.

T	s starting time	τ stopping time	η strength
28	0	6 , 7, 8	0.9 , 1.0

For FlowChef, the paper provides a set of hyperparameters for each kind of editing. Moreover, in the official implementation for each image, there is a different set of hyperparameters, and these hyperparameters are different from the ones provided in the paper. Therefore, we evaluate all the sets provided in their paper and in the official implementation, as well as several other sets. We report the sets that achieved the best results in Tab. S5. In all our comparisons we used the set that achieved the best results. We note that FlowChef’s method can accept a user-provided mask, which may optionally be constant over the entire image. Since the official implementation lacks the annotation-free editing option, which automatically determines a mask by itself, we follow the gradio implementation and use a constant mask.

Table S5: FlowChef hyperparameters.

T	Max steps	CFG	Full source steps	η strength	Optimization steps
30	20	4.5	5	0.4, 0.5	1, 2 , 3, 4

We had also evaluated other sets of hyperparameters that had not achieved better results than those reported. For additional comparisons, see App. B.2.5.

For FLUX Kontext, we evaluate the default set of hyperparameters as well as another set, report them in Tab. S6 and choose the set that achieved the best results.

Table S6: FLUX Kontext hyperparameters.

T	CFG
28	2.5 , 3.5

B.2.3 USER STUDY

We selected 30 random triplets of a source image, a source prompt and target prompt from our dataset, and constructed a user study of pairwise comparisons between the editing results achieved by our method against three competing methods. An example of a pairwise comparison is provided in Fig. S5. For each pairwise comparison, the user was asked three forced two-alternative questions about the preferred editing result (image A or image B): which is more similar to the source image? which better adheres to the target text? and which editing result is preferable overall?

Pink, red and white flowers → Orange, yellow and white flowers
Reference **A** **B**



Figure S5: Pairwise comparison example from our user study.

B.2.4 COMPARISON TO FLUX KONTEXT

FLUX Kontext (Black Forest Labs et al., 2025) is a SotA editing model, that achieves its impressive results by fine-tuning FLUX base model (Black Forest Labs, 2024). Although being fine-tuned,

zero-shot methods, and specifically FlowOpt, achieve highly competitive results compare to FLUX Kontext in both text adherence and structure preserving metrics, as illustrated in Figs. 7 and S4.

Yet, the main advantage of FlowOpt compared to FLUX Kontext, and any other trained image editing method, is the fact that this method can be easily applied to any new text-to-image flow/diffusion model, such as future releases of Stable Diffusion and FLUX, without requiring expensive and resource intensive training.

Naturally, trained methods have advantages compared to zero-shot methods. Mainly, they can achieve edits that are highly challenging for zero-shot structure-preserving methods such as FlowOpt, including changing the color palate of the source image and changing the pose of the objects in the image. However, for replacing the main object in the image or changing its style, FlowOpt achieves highly comparable, and in some cases better, results compared to FLUX Kontext. See Fig. S6 for qualitative comparisons.

FLUX Kontext was trained for editing using instructions, *e.g.*, instead of using the target prompt “A large tiger standing in a swamp” for an image described by the source prompt “A large tiger standing in a swamp”, its input should be “Change the tiger to a crochet tiger”. Therefore, to compare it to FlowOpt using our dataset we used Gemini 2.5 (Comanici et al., 2025) to modify our target prompts to instructions and manually refined them.

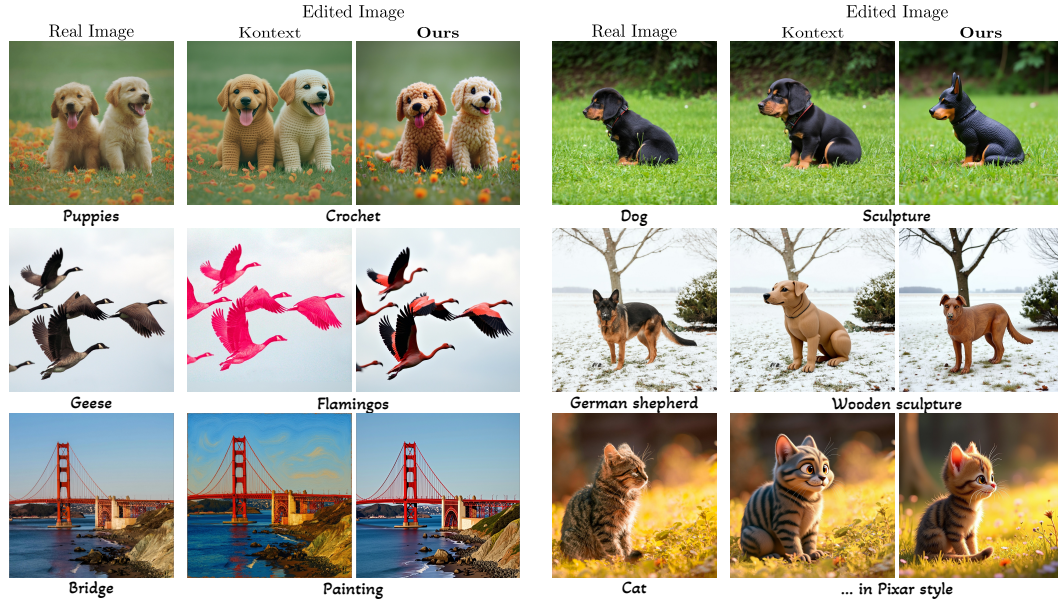


Figure S6: Qualitative comparison between FlowOpt (FLUX) and FLUX Kontext. Fine details are visible upon zooming in. In the first and second rows, for replacing the main object in the scene, FlowOpt achieves better results compared to FLUX Kontext. Finally, In the last row, FLUX Kontext achieves better editing results, as its capacity for changing the color palette of the source image is larger.

B.2.5 COMPARISON TO FLOWCHEF

Consistent with the official implementation of FlowChef and the hyperparameters specified in their paper, all evaluated configurations are detailed in Tab. S7. A detailed quantitative comparison between FlowChef and FlowOpt is provided in Fig. S7. The alignment of the text is measured using the CLIP-Text measure, and the fidelity of the image is measured using CLIP-Image.

Table S7: Detailed FlowChef hyperparameters evaluated.

T	Max steps	CFG	Full source steps	η strength	Optimization steps
30	20	4.5	0	0.5	1, 2, 3, 4, 5
30	20	4.5	2	0.5	1, 2, 3, 4, 5
30	20	4.5	5	0.4, 0.5	1, 2, 3, 4, 5

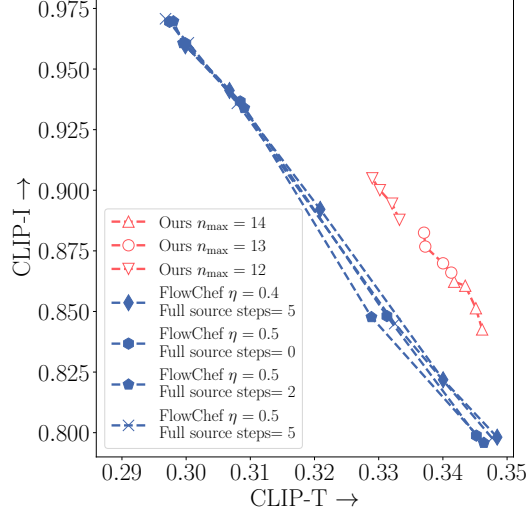


Figure S7: Detailed editing quantitative comparisons (FLUX) between FlowChef and FlowOpt. Text adherence is measured by CLIP-Text (x-axis), and image adherence (y-axis) is measured by CLIP-Image. Connected markers represent different hyperparameters.

C INITIALIZATION

We proved that if the step size is chosen appropriately, then FlowOpt necessarily converges to the unique global minimum of our optimization problem. However, for any finite number of iterations, the initialization does have an impact on the result. This is illustrated in Figs. S8 and S10, where the red and yellow curves correspond to initialization with the UniInv and ODE Inversion methods, respectively. As the results obtained with the UniInv initialization are better than with ODE inversion, we chose the former for all experiments in the paper.

We further compare the impact of the aforementioned initializations (UniInv and ODE Inversion) to an initialization with a sample of white Gaussian noise. Figure S9 illustrates the convergence obtained for these different initializations on 5 different random images from the DIV2K dataset. Note that the x-axis appears in logarithmic scale. We can see that, as predicted by Theorem 1, our method converges also with a random initialization when using the same value of η that satisfies the condition of the theorem, namely $\eta = 2.5 \cdot 10^{-3}$.

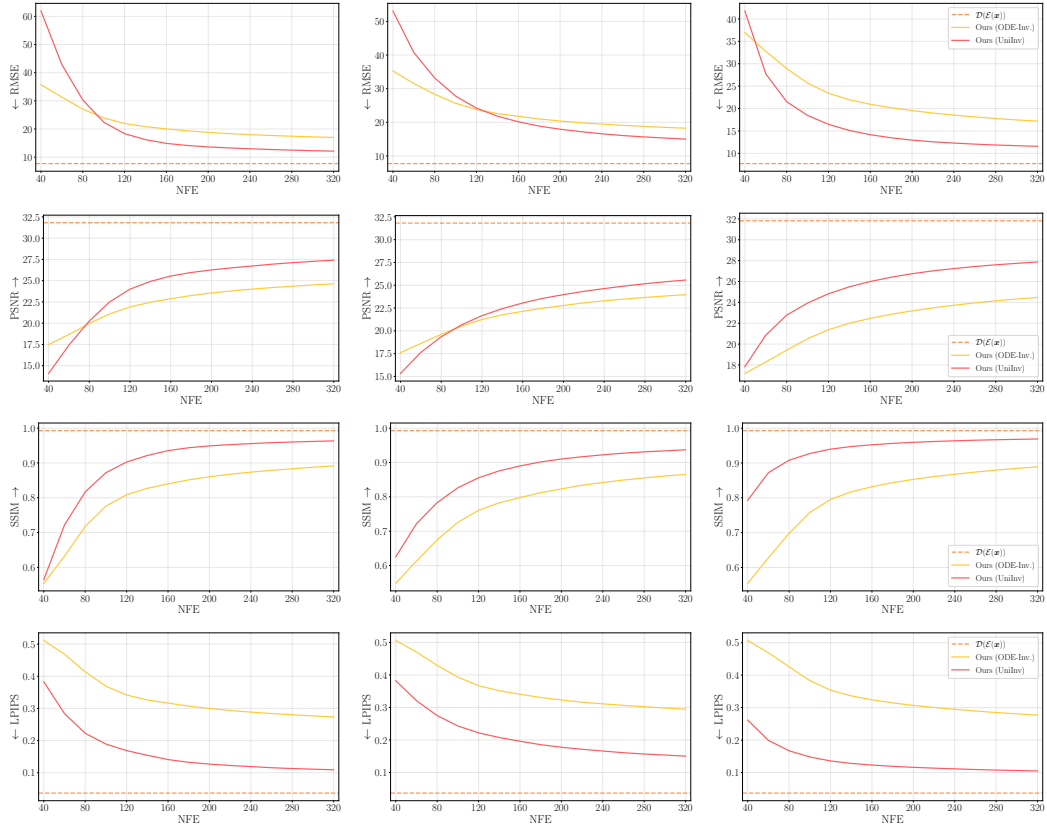


Figure S8: Reconstruction quantitative comparisons (FLUX). Pixel-space RMSE (first row), PSNR (second row), SSIM (third row), and LPIPS (last row) as a function of the number of NFEs, for unconditional sampling with CFG = 1 (left), unconditional sampling with CFG = 0 (center), and text-conditional sampling (right). The red and yellow curves correspond to FlowOpt initialized with the UniInv and ODE Inversion methods, respectively. The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model.

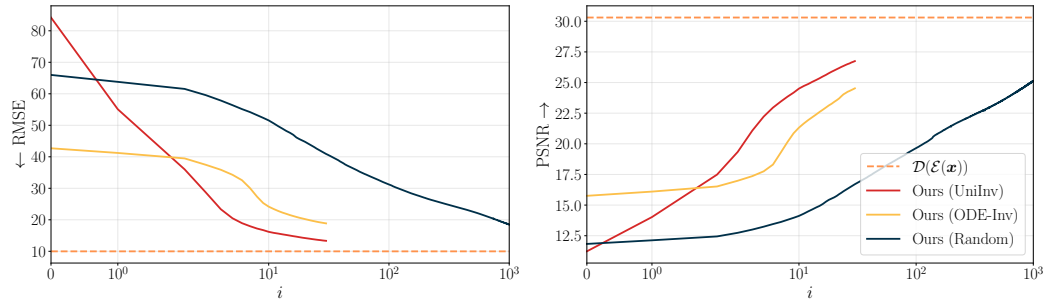


Figure S9: Reconstruction initialization quantitative comparisons (FLUX). Pixel-space RMSE (left) and PSNR (right) as a function of the number of iterations in logarithmic scale, for text-conditional sampling. The red and yellow curves correspond to FlowOpt initialized with the UniInv and ODE Inversion methods, as well as random white Gaussian noise initialization, respectively. The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model.

D STABLE DIFFUSION 3 (SD3)

In this appendix, we repeat all the experiments in Sec. 5, but with SD3 instead of FLUX. In this case, we choose the step size $\eta = 10^{-2}$ in the update rule of Eq. (6).

D.1 IMAGE RECONSTRUCTION (INVERSION)

Implementation details. We use the implementation details provided for FLUX. We set the number of denoisers to $T = 10$, and evaluate the reconstruction error for various NFE values.

Dataset. We use the same dataset used for evaluating FLUX – randomly chosen same 100 real images of dimension 1024×1024 from the DIV2K dataset, that was automatically captioned by BLIP, and manually refined.

Competing methods. As with the experiments on FLUX, we compare our method to ODE Inversion, RF-Solver, FireFlow and UniInv. For methods, like RF-Solver, which use two forward passes per timestep, we set $T = \frac{\text{NFE}}{4}$. For methods that use a single forward pass per timestep, we set $T = \frac{\text{NFE}}{2}$. We also evaluated ReNoise, with both $T = \{10, 28\}$, and set the number of ReNoise steps so as to achieve the desired NFE count. We evaluated various hyperparameters for ReNoise and report the results with those that worked best. It should be noted that, for the fixed point iterations of ReNoise, one would expect that the final iteration would provide the best results. However, we observe that this does not necessarily happen in practice. We also note that as there was no official implementation for any method for SD3, we implemented all of them by ourselves.

Quantitative evaluation. The reconstruction results of FlowOpt, as well as competing the methods are provided in Fig. S10 both for the unconditional and the conditional case. We can see that our method achieves the best reconstruction results for various NFE values, both for unconditional and for conditional sampling. We can also see that the initialization affects the results achieved by our method, with UniInv leading to better results than naive ODE Inversion and outperforming the competing methods.

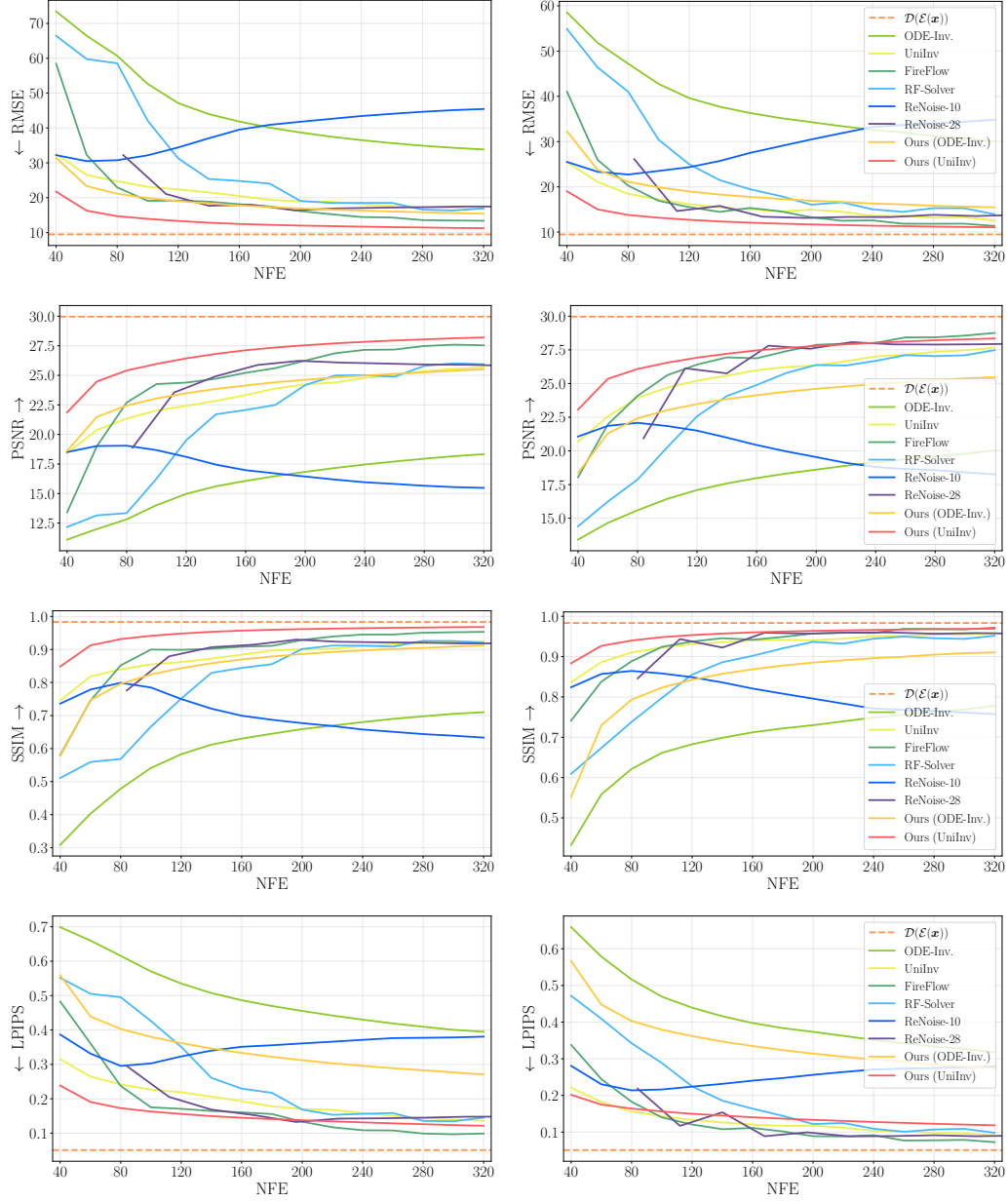


Figure S10: Reconstruction quantitative comparisons (SD3). Pixel-space RMSE (first row), PSNR (second row), SSIM (third row), and LPIPS (last row) as a function of the number of NFEs for several inversion methods, for unconditional (left) and text-conditional (right) sampling. The red and yellow curves correspond to our FlowOpt initialized with the UniInv and ODE Inversion methods, respectively. The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model.

D.2 IMAGE EDITING

Implementation details. We set the number of denoisers to $T = 15$, and evaluate our method for various n_{\max} values. Specifically, we use $n_{\max} \in \{13, 12\}$. We set the CFG to the default value, *i.e.*, $\text{CFG} = 3.5$. We evaluate our method for various number of iterations, $N \in \{2, 3, 4, 5\}$. For all figures we present the results obtained with $n_{\max} = 12$.

Dataset. We use the same dataset used for evaluating FLUX – about 400 text-image pairs. The dataset consists of 90 real images of dimensions 1024×1024 , which were captioned by LLaVA-1.5, and manually refined. The target prompts for editing the images were handcrafted.

Competing methods. We compare our method against ODE Inversion, UniEdit and FlowEdit. As there was no official implementation for UniEdit for SD3, we implemented it by ourselves. For ODE Inversion, we apply the same number of NFEs used for our method. For all methods, we performed hyperparameters search. Additional details regarding the hyperparameters are provided below.

Quantitative evaluation. We evaluate the results of all methods using the same measures reported for FLUX in Sec. 5. The results are presented in Fig. S11. We see that our method achieves results comparable to FlowEdit, and achieves better results than other competing methods.

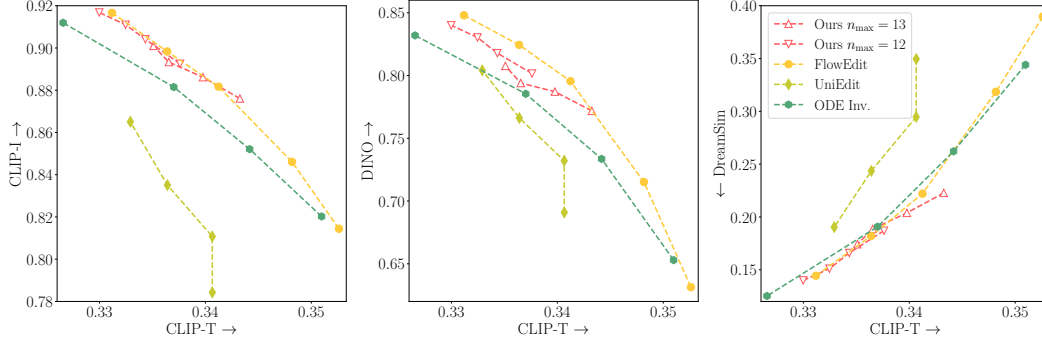


Figure S11: Editing quantitative comparisons (SD3). Text adherence is measured by CLIP-Text (x-axis) for all figures. Image adherence (y-axis) is measured by CLIP-Image (left), DINOv3 (center), and DreamSim (right). Connected markers represent different hyperparameters.

Qualitative evaluation. Figure S12 shows comparisons between FlowOpt method and the competing methods. More details about the hyperparameters used to construct this figure are provided in Sec. D.2.1. We can see that our method achieves at least comparable results to other methods, for both object editing and style editing. For example, FlowOpt is the only method that preserves the structure of the scene and the running kid (second row), and successfully turns him into a sculpture. Moreover, our method is the only one that preserves the cat and the crown structure (fifth row), and successfully edits its color. Additional results of our method are provided in Fig. S13.

594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647



Figure S12: Qualitative comparisons (SD3). Fine details are visible upon zooming in.

648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

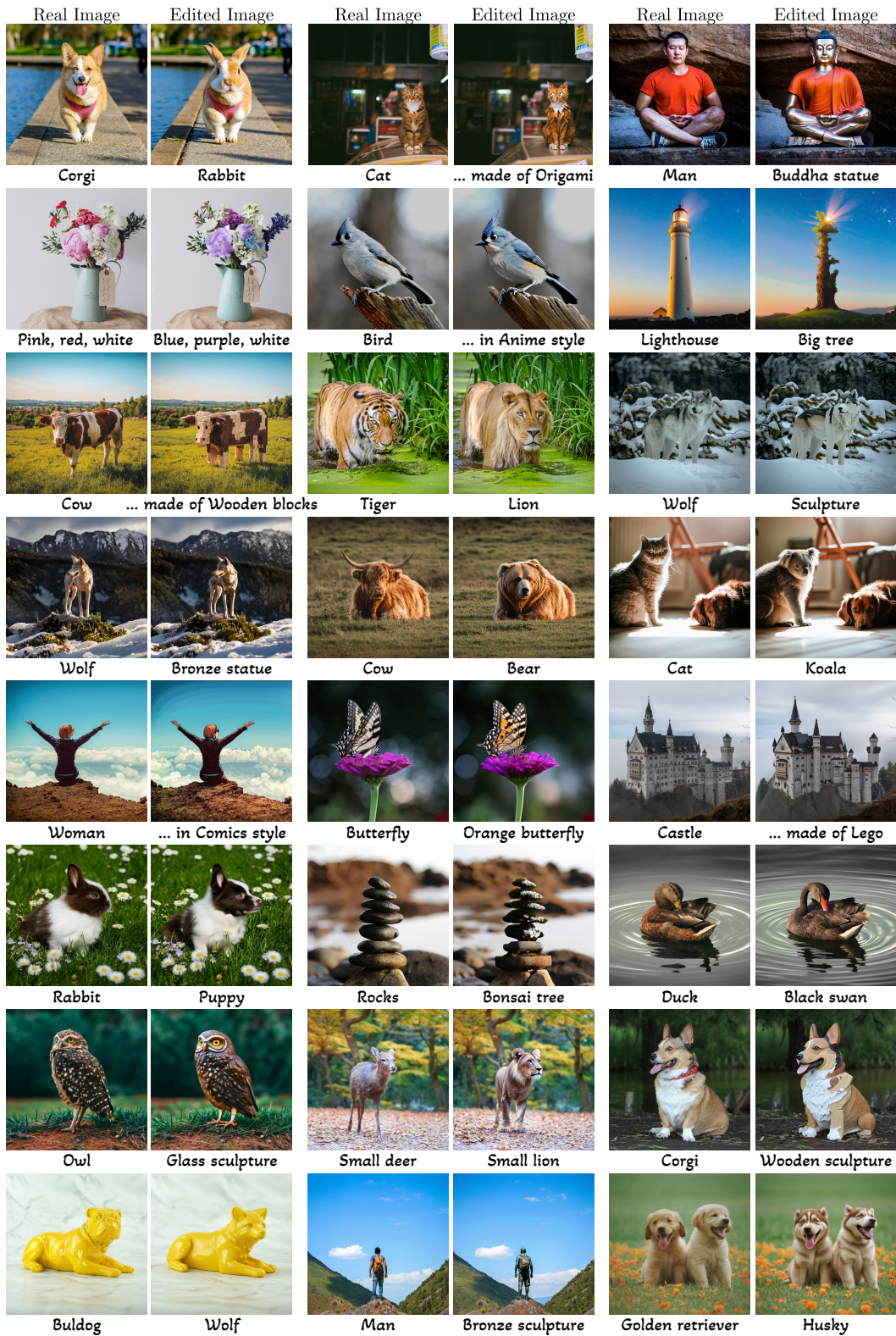


Figure S13: Additional FlowOpt results (SD3). Fine details are visible upon zooming in.

D.2.1 SD3 HYPERPARAMETERS.

The hyperparameters presented in Fig. S12 for FlowEdit, ODE Inversion, and FlowOpt are listed in Tab. S8, where the chosen hyperparameters for the displayed figures are marked in bold.

Table S8: SD3 hyperparameters.

	T	n_{\max}	CFG @ source	CFG @ target	N iterations
FlowEdit	50	45, 40, 33 , 30, 27	3.5	13.5	-
ODE Inversion	50	45, 40 , 35, 30	1	3.5	-
FlowOpt	15	13, 12	1	3.5	2, 3, 4, 5

For UniEdit, the evaluated hyperparameters are presented in Tab. S9, with the chosen value for their α parameter marked in bold. In our notation, $\alpha = n_{\max}/T$.

Table S9: SD3 UniEdit hyperparameters.

T	α delay rate	ω guidance scale
15	$\frac{2}{5}$, $\frac{2}{3}$, $\frac{11}{15}$, $\frac{3}{5}$	5

E EDITING WITH OTHER LOSS FUNCTIONS

As noted in Sec. 4, we can generalize the MSE loss defined in Eq. (4) to other loss functions $\mathcal{L}(f(z_t), y)$. In this case, the update rule in Eq. (6) becomes

$$z_t^{(i+1)} \leftarrow z_t^{(i)} - \eta \nabla_f \mathcal{L}(f(z_t^{(i)}), y). \quad (\text{S1})$$

Note that Eq. (S1) uses the gradient of the loss, but not the Jacobian of f . That is, it does not require backpropagating through the flow process, though it does require backpropagating through decoder in cases where the loss is defined in pixel-space. However, while seemingly attractive, we have not seen significant advantages for using losses other than the L^2 loss, except of infrequent cases, as presented in Fig. S16 (for FLUX). Moreover, we observed that other losses typically achieved satisfying results for larger number of iterations (N). Specifically, other losses typically require $\sim 15 - 30$ iterations, which is significantly more than the $\sim 3 - 5$ iterations that commonly suffice for the L^2 loss. Therefore, the L^2 loss has the advantage of achieving satisfying results while being computationally. Figures S14, S15 present results with different loss functions, for both SD3 and FLUX. These include the contextual loss (CX) (Mechrez et al., 2018) in pixel space and the ELatentLPIPS (Kang et al., 2024) in latent space, in addition to our default latent-space L^2 loss. For all loss functions, we used the hyperparameters reported in Sec. 5 for FLUX, and in App. D for SD3, except for N and η . We can see that the CX and ELatentLpips losses achieve similar results to the ones obtained with the L^2 loss.

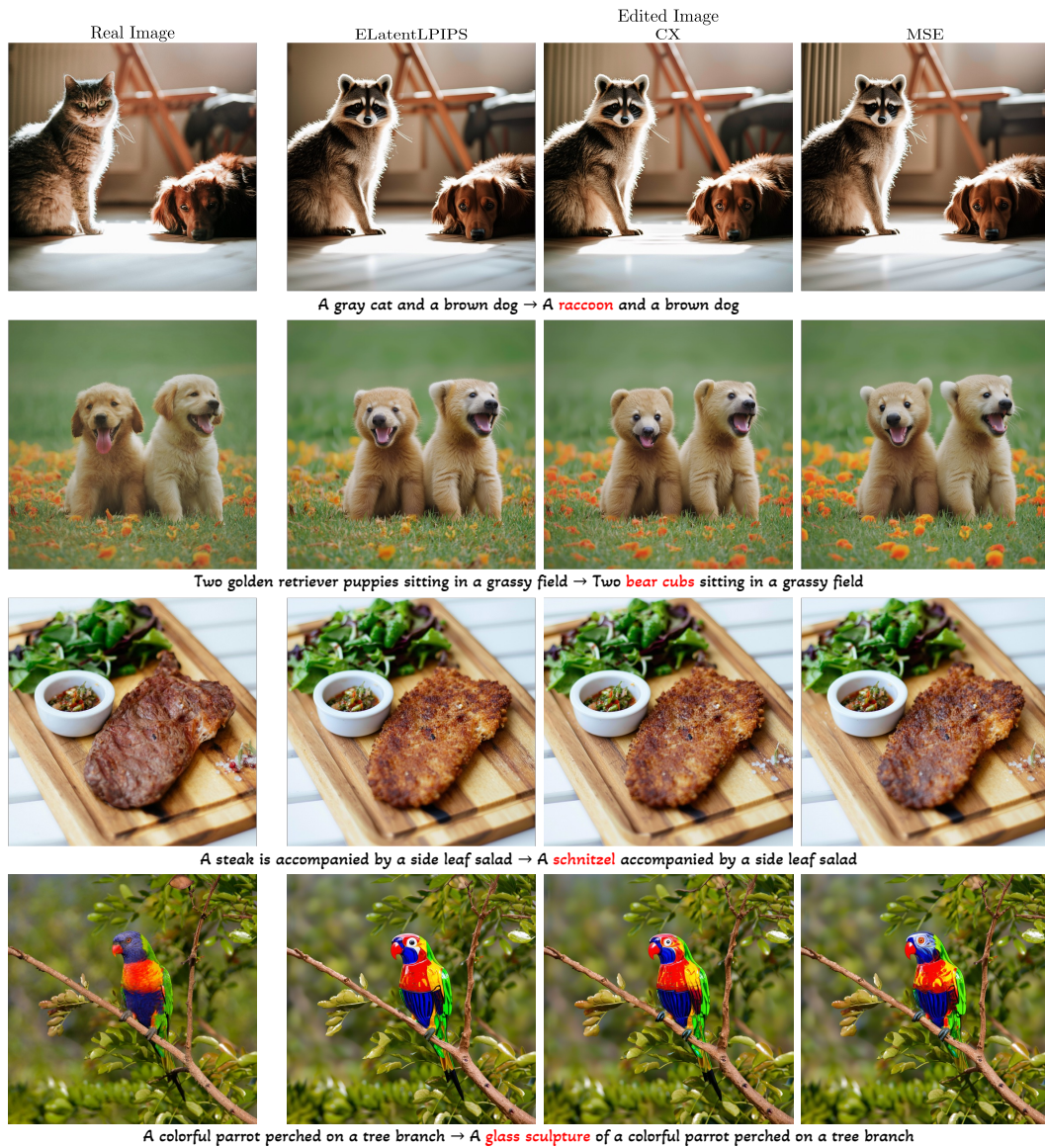


Figure S14: Qualitative comparisons using other loss functions (SD3). The results obtained for the update rule in Eq. (S1), for ELatentLPIPS loss (left), contextual (CX) loss (center), and our proposed approach – MSE loss (right). The results obtained by all losses are similar.



Figure S15: Qualitative comparisons using other loss functions (FLUX). The results obtained for the update rule in Eq. (S1), for ELatentLPIPS loss (left), contextual (CX) loss (center), and our proposed approach – MSE loss (right). The results obtained by all losses are similar.

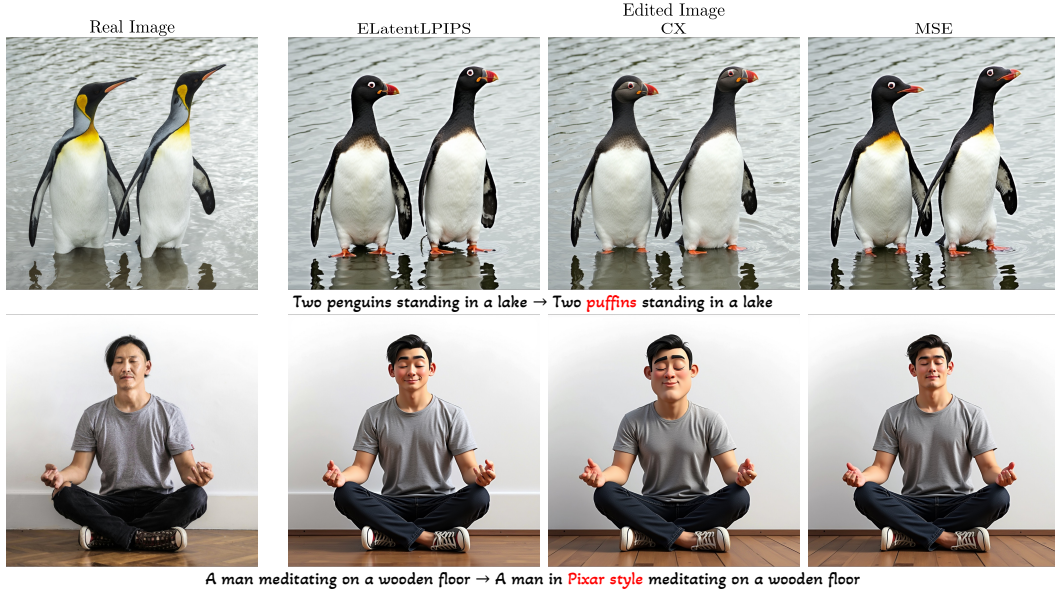


Figure S16: Qualitative comparisons using other loss functions (FLUX). The results obtained for the update rule in Eq. (S1), for ELatentLPIPS loss (left), contextual (CX) loss (center), and our proposed approach – MSE loss (right). Infrequent cases where the results obtained with loss functions other than MSE are better than results obtained with the MSE loss (allowing weaker structure preservation in favor of stronger adherence to the target text prompt). The number of iterations is typically larger for other loss functions.

F CONTRACTION MAPPING

F.1 PROOF OF THEOREM 1

Let $g(\mathbf{u}) \triangleq \mathbf{u} - \eta(f(\mathbf{u}) - \mathbf{y})$. By definition, $g(\mathbf{u})$ is a contraction mapping if there exists $\gamma \in [0, 1)$ such that

$$\|g(\mathbf{u}_1) - g(\mathbf{u}_2)\| \leq \gamma \|\mathbf{u}_1 - \mathbf{u}_2\| \quad (\text{S2})$$

for all $\mathbf{u}_1, \mathbf{u}_2$. Substituting g , the inequality reads

$$\|(\mathbf{u}_1 - \eta(f(\mathbf{u}_1) - \mathbf{y})) - (\mathbf{u}_2 - \eta(f(\mathbf{u}_2) - \mathbf{y}))\| \leq \gamma \|\mathbf{u}_1 - \mathbf{u}_2\|. \quad (\text{S3})$$

Squaring both sides, we get

$$\|\mathbf{u}_1 - \mathbf{u}_2 - \eta(f(\mathbf{u}_1) - f(\mathbf{u}_2))\|^2 \leq \gamma^2 \|\mathbf{u}_1 - \mathbf{u}_2\|^2. \quad (\text{S4})$$

Rearranging terms, we get

$$\|\mathbf{u}_1 - \mathbf{u}_2\|^2 + \eta^2 \|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2 - 2\eta \langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle \leq \gamma^2 \|\mathbf{u}_1 - \mathbf{u}_2\|^2. \quad (\text{S5})$$

Defining $\kappa = 1 - \gamma^2 \in (0, 1]$, we get a quadratic inequality in η ,

$$\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2 \eta^2 - 2 \langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle \eta + \kappa \|\mathbf{u}_1 - \mathbf{u}_2\|^2 \leq 0. \quad (\text{S6})$$

For each given pair of $\mathbf{u}_1, \mathbf{u}_2$, the set of η 's that satisfy the inequality is $\eta \in [\eta_1(\mathbf{u}_1, \mathbf{u}_2), \eta_2(\mathbf{u}_1, \mathbf{u}_2)]$, where

$$\begin{aligned} \eta_{1,2}(\mathbf{u}_1, \mathbf{u}_2) &= \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \pm \sqrt{\left(\frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \right)^2 - \kappa \left(\frac{\|\mathbf{u}_1 - \mathbf{u}_2\|}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|} \right)^2} \\ &= \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \left(1 \pm \sqrt{1 - \kappa \left(\frac{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\| \|\mathbf{u}_1 - \mathbf{u}_2\|}{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle} \right)^2} \right). \end{aligned} \quad (\text{S7})$$

Therefore, if we choose

$$\eta \in (\bar{\eta}_1, \bar{\eta}_2) \subset \left[\sup_{\mathbf{u}_1, \mathbf{u}_2} \eta_1(\mathbf{u}_1, \mathbf{u}_2), \inf_{\mathbf{u}_1, \mathbf{u}_2} \eta_2(\mathbf{u}_1, \mathbf{u}_2) \right], \quad (\text{S8})$$

then the iterations are guaranteed to converge. To choose $\bar{\eta}_2$, we note that

$$\begin{aligned} & \inf_{\mathbf{u}_1, \mathbf{u}_2} \eta_2(\mathbf{u}_1, \mathbf{u}_2) \\ & \geq \inf_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \inf_{\mathbf{u}_1, \mathbf{u}_2} \left(1 + \sqrt{1 - \kappa \left(\frac{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\| \|\mathbf{u}_1 - \mathbf{u}_2\|}{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle} \right)^2} \right) \\ & = \inf_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \left(1 + \sqrt{1 - \kappa \sup_{\mathbf{u}_1, \mathbf{u}_2} \left(\frac{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\| \|\mathbf{u}_1 - \mathbf{u}_2\|}{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle} \right)^2} \right) \\ & \geq \inf_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \left(1 + \sqrt{1 - \frac{\kappa}{\beta^2}} \right) \\ & \triangleq \bar{\eta}_2, \end{aligned} \quad (\text{S9})$$

where we denoted $\beta = \inf_{\mathbf{u}_1 \neq \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|\mathbf{u}_1 - \mathbf{u}_2\| \|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|}$ and used the assumption of the theorem that $\beta > 0$. Note that the first inequality here follows from the fact that both multiplicands are nonnegative, as $\inf_{\mathbf{u}_1 \neq \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} > 0$ from the assumption of Eq. (7) in the theorem. In a similar manner, we can choose $\bar{\eta}_1$ by noting that

$$\sup_{\mathbf{u}_1, \mathbf{u}_2} \eta_1(\mathbf{u}_1, \mathbf{u}_2) \leq \sup_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} \left(1 - \sqrt{1 - \frac{\kappa}{\beta^2}} \right) \triangleq \bar{\eta}_1. \quad (\text{S10})$$

Now, since $\kappa > 0$ can be chosen arbitrarily small, we take the upper bound to be

$$\lim_{\kappa \rightarrow 0} \bar{\eta}_2 = 2 \inf_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2}, \quad (\text{S11})$$

and since $\sup_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} < \infty$, we take the lower bound to be

$$\lim_{\kappa \rightarrow 0} \bar{\eta}_1 = 0. \quad (\text{S12})$$

This is allowed since for any $\eta \in (\lim_{\kappa \rightarrow 0} \bar{\eta}_1, \lim_{\kappa \rightarrow 0} \bar{\eta}_2)$, there exists a fixed $\kappa > 0$ small enough such that Eq. (S8) is satisfied with that particular κ . This completes the proof of the theorem.

F.2 STEP SIZE UPPER BOUND

To verify our choice of η , we drew many pairs of samples $\mathbf{u}_1, \mathbf{u}_2$, as we detail next. Specifically, we generated nonidentical 2000 text prompts using ChatGPT4 (Achiam et al., 2023), drew two different random white Gaussian noises $\mathbf{u}_1, \varepsilon$ for each text prompt, and defined \mathbf{u}_2 as

$$\mathbf{u}_2 = \sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon, \quad (\text{S13})$$

for various α values, so that both \mathbf{u}_1 and \mathbf{u}_2 are distributed $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (an isotropic Gaussian).

By substituting \mathbf{u}_2 into Eq. (S11) we obtain, for each α value

$$\min_{\mathbf{u}_1, \varepsilon} 2 \frac{\left\langle (1 - \sqrt{\alpha}) \mathbf{u}_1 - \sqrt{1 - \alpha} \varepsilon, f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon) \right\rangle}{\|f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon)\|^2}. \quad (\text{S14})$$

Figure S17 presents the minimum in Eq. (S14) obtained over the 2000 different realizations, as a function of α , for both FLUX and SD3. The global minimum, marked by a blue star, is our approximation for the upper bound of Eq. (7). We can see that the minimum is obtained when $\|\mathbf{u}_1 - \mathbf{u}_2\|$ is small (α close to 1). Our choice for η , which is presented as a dashed red line, is below this upper bound.

Figure S18 is the same as Fig. 5, but for FLUX instead of SD3, and the comparisons are to step sizes that are $4\times$ and $10\times$ larger than our choice, namely $\eta \in \{1.0 \times 10^{-2}, 2.5 \cdot 10^{-2}\}$.

We note that this experiment was evaluated for $T = 10$ both for SD3 and FLUX, and for latent variables $\{z_t\}$ corresponding to images with dimensions of 1024×1024 . For a different number of denoisers, or images of other resolutions, the experiment should be redone.

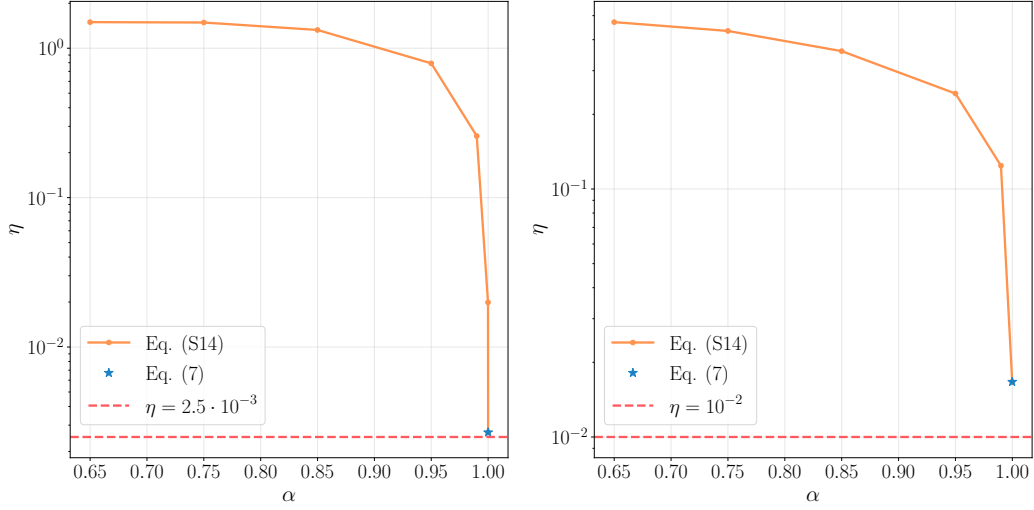


Figure S17: Step size upper bounds. The orange line is the minimum obtained over 2000 noise realizations in Eq. (S14), achieved for various α values. The approximation for the upper bound (Eq. (7)) is the starred blue point, and the dashed red line is our step size (η) choice (which is below the upper bound), for FLUX (left) and SD3 (right).

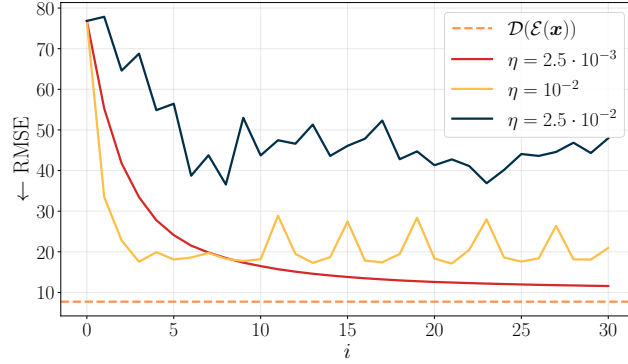


Figure S18: Convergence analysis (FLUX). The plot shows RMSE in pixel space vs. number of iterations for the task of inversion, averaged over a dataset. The step size we use (red) satisfies the sufficient condition of Eq. (7) and thus leads to convergence. Step sizes that are $4\times$ and $10\times$ larger (yellow and black) do not satisfy the condition and do not lead to convergence. The dashed orange line is the minimal RMSE achievable in this setting. It corresponds to passing images through the encoder and decoder.

F.3 VALIDATION OF THE ASSUMPTIONS OF THEOREM 1

To validate the assumptions of theorem 1, we conducted a similar experiment to that detailed in App. F.2, but with 1000 text prompts instead of 2000.

By substituting \mathbf{u}_2 into the left-hand side of the assumption $\inf_{\mathbf{u}_1 \neq \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|\mathbf{u}_1 - \mathbf{u}_2\| \|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|} > 0$ we obtain, for each α value

$$\min_{\mathbf{u}_1, \varepsilon} \frac{\left\langle (1 - \sqrt{\alpha}) \mathbf{u}_1 - \sqrt{1 - \alpha} \varepsilon, f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon) \right\rangle}{\|(1 - \sqrt{\alpha}) \mathbf{u}_1 - \sqrt{1 - \alpha} \varepsilon\| \|f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon)\|}, \quad (\text{S15})$$

and by substituting \mathbf{u}_2 into the left-hand side of the assumption $\sup_{\mathbf{u}_1, \mathbf{u}_2} \frac{\langle \mathbf{u}_1 - \mathbf{u}_2, f(\mathbf{u}_1) - f(\mathbf{u}_2) \rangle}{\|f(\mathbf{u}_1) - f(\mathbf{u}_2)\|^2} < \infty$ we obtain, for each α value

$$\max_{\mathbf{u}_1, \varepsilon} \frac{\left\langle (1 - \sqrt{\alpha}) \mathbf{u}_1 - \sqrt{1 - \alpha} \varepsilon, f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon) \right\rangle}{\|f(\mathbf{u}_1) - f(\sqrt{\alpha} \mathbf{u}_1 + \sqrt{1 - \alpha} \varepsilon)\|^2}. \quad (\text{S16})$$

As illustrated in Fig. S19, our evaluations confirm that these conditions hold. Specifically, the lower bound in Eq. (S15) remains strictly positive, finite, and bounded (left). Additionally, the upper bound in Eq. (S16) remains finite (right). Moreover, Fig. S17 demonstrates that the strict positivity assumption for Eq. (7) is also satisfied – for \mathbf{u}_1 and \mathbf{u}_2 that are farther apart, the left-hand side is larger than when they are closer. Note that both axes in Fig. S19 are in logarithmic scale.

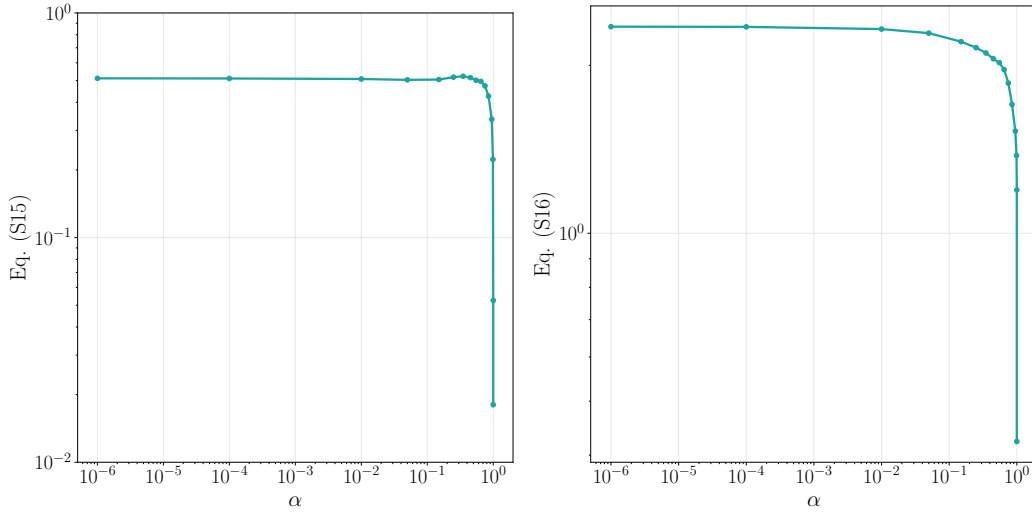


Figure S19: Theorem 1 assumptions validation (FLUX). The minimum obtained over 1000 noise realizations in Eq. (S15), achieved for various α values (left). Similarly, the maximum obtained in Eq. (S16) for the same setting (right). Both x- and y-axes are plotted on a logarithmic scale.

G PROBABILITY FLOW ODE COEFFICIENTS

Each denoising step by the flow formulation is given by

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \mathbf{v}_t(\mathbf{z}_t)\Delta t, \quad (\text{S17})$$

where for notational convenience we omit the condition c .

However, each denoising step by the DDIM formulation is given by

$$\mathbf{z}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{z}_t - \sqrt{1 - \alpha_t} \epsilon_\theta^t(\mathbf{z}_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta^t(\mathbf{z}_t), \quad (\text{S18})$$

where α_t are the diffusion coefficients as defined by Song et al. (2021a), and $\epsilon_\theta^t(\mathbf{z}_t)$ is the predicted noise for the current observation \mathbf{z}_t , replacing the learned vector field $\mathbf{v}_t(\mathbf{z}_t)$ of the flow formulation. Rearranging Eq. (S18), we get

$$\mathbf{z}_{t-1} = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} \mathbf{z}_t + \left(\sqrt{1 - \alpha_{t-1}} - \frac{\sqrt{1 - \alpha_t}}{\sqrt{\alpha_t}} \right) \epsilon_\theta^t(\mathbf{z}_t). \quad (\text{S19})$$

As we relate to the entire process as a black box, and a `stop-grad` operator is applied on the output of each of the noise-predicting networks, the terms $\epsilon_\theta^t(\mathbf{z}_t)$ vanish under differentiation. Stacking all timesteps one after the other, the formulation remains the same as flows, but with a multiplicative coefficient that corresponds to the product of the coefficients multiplying \mathbf{z}_t in each of the timesteps,

$$\delta \triangleq \prod_{t=1}^T \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} = \sqrt{\frac{\alpha_0}{\alpha_T}} = \frac{1}{\sqrt{\alpha_T}}. \quad (\text{S20})$$

Therefore, for example, the update rule for the L^2 loss in Eq. (4) for any condition c , is given by

$$\mathbf{z}_t^{(i+1)} \leftarrow \mathbf{z}_t^{(i)} - \eta \delta \left(f(\mathbf{z}_t^{(i)}, c) - \mathbf{y} \right). \quad (\text{S21})$$

H HYPERPARAMETERS USED FOR FIGURE 1

The results presented in Fig. 1 were achieved by the hyperparameters provided in Tab. S10.

Table S10: Figure 1 hyperparameters.

	Model	n_{\max}	N iterations
Owls \rightarrow Cardboard	FLUX	11	5
Corgi \rightarrow Lego	FLUX	13	8
Forest \rightarrow Paved pathway	FLUX	13	3
Penguins \rightarrow Glass sculpture	SD3	12	4
Owl \rightarrow in Anime style	SD3	12	5
Wolf \rightarrow Deer	SD3	12	4
Cow \rightarrow Colorful toy bricks	FLUX	12	6
Lizard \rightarrow Crochet	FLUX	12	5
Corgi \rightarrow in Pixar style	FLUX	11	5

I EDITING BY INVERSION

In this section we show that even if we have a good inversion method, we do not necessarily get good editing performance with the naive editing-by-inversion paradigm. Our definition for good inversion is that we have a noise map z_t , such that if we would forward it through the chain of denoisers, $f(z_t, c_{\text{src}})$, we would get almost the same original image z_0 . Figure S20 demonstrates this using the inversion map obtained by Eq. (6) with $c = c_{\text{src}}$, where in the final iteration we perform sampling with $c = c_{\text{tar}}$. As can be seen, while increasing the number of iterations N leads to better inversion (top row), it does not monotonically improve the editing measures (bottom plots). Here, we used $T = 15$ and $n_{\text{max}} = 13$ with the same dataset as that we used for the image editing experiment of Sec. 5.

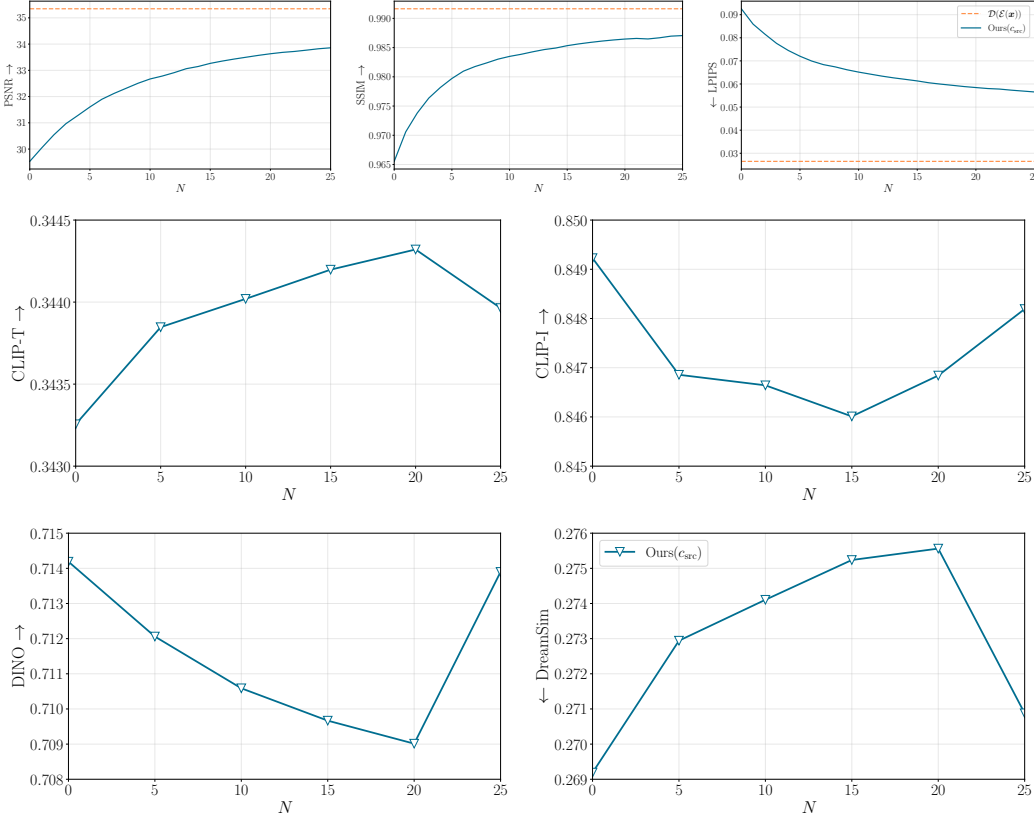


Figure S20: Editing by inversion (FLUX). Reconstruction metrics (top) – pixel-space PSNR, SSIM and LPIPS (left to right), and editing metrics (last two rows) – CLIP-Text, CLIP-Image, DINOv3 and DreamSim (left to right, top to bottom), as a function of the number of iterations (N) by using $c = c_{\text{src}}$ in Eq. (4), and in the final sampling step using $c = c_{\text{tar}}$. The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model. Better Inversion (first row) does not imply better editing – no improvement trend is observed in the editing metrics, even as reconstruction quality improves.

J CONVERGENCE OF FLOWOPT FOR OTHER DIMENSIONS

In this section, we show the convergence of FlowOpt for additional dimensions than 1024×1024 , specifically here we show the convergence for images with dimensions of 512×512 and 256×256 . We note that we used $\eta = 1 \cdot 10^{-2}$ for images with dimensions of 512×512 and $\eta = 2 \cdot 10^{-2}$ for images with dimensions of 256×256 , since according to Theorem 1 η depends on $f(\cdot)$, where $f(\cdot)$ depends on the dimensions.

For evaluation, we repeat the experiment of Sec. 5.1 for text-conditional sampling. Figure S21 demonstrates that our method converges for different dimensions, where the first row corresponds to images with dimensions of 1024×1024 , the second row corresponds to images with dimensions of 512×512 , and the last row corresponds to images with dimensions of 256×256 . We can clearly see that for all dimensions FlowOpt converges rapidly. We note that this boundary is different for different dimensions.

We note that in general, we rely on the Banach fixed point theorem (Banach, 1922), where the convergence is in the geometric sense and the convergence rate depends on the contraction constant (γ in App. F). This constant depends on the dimension as well as other factors.

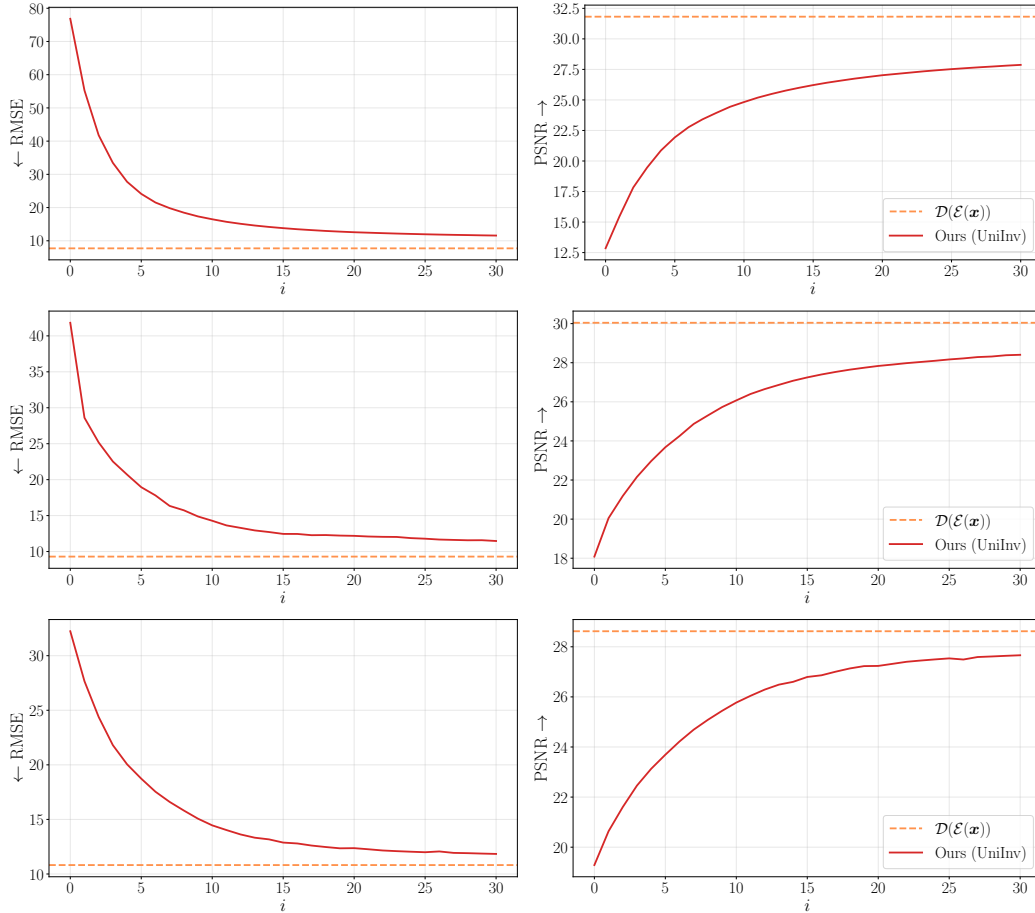


Figure S21: Reconstruction quantitative evaluation for different images' dimensions (FLUX). Pixel-space RMSE (left) and PSNR (right) as a function of the number of iterations, for text-conditional sampling, for images with dimensions of 1024×1024 (first row), 512×512 (second row) and 256×256 (last row). The red curve corresponds to FlowOpt initialized with the UniInv method. The dashed orange horizontal line is the average of forwarding the images through the encoder and decoder of the model.

K FURTHER DISCUSSION ON THE RELATION TO FLOWCHEF METHOD

As mentioned in Sec. 2, FlowOpt could superficially seem similar to FlowChef (Patel et al., 2025), as both are zero-order optimization methods for flow process. However, there is a fundamental difference between the two methods. During the denoising (sampling) process, FlowChef optimizes each intermediate latent z_t separately, attempting to draw the prediction $\hat{z}_0 = z_t - v_t(z_t, c)t$ closer to the reference image y . This is done via a simple guidance step. In sharp contrast, FlowOpt optimizes over the initial noise z_1 (or optionally over z_{t_0} for some fixed timestep t_0) and treats the entire flow process as a black box. As a result, FlowChef does not construct an initial noise z_1 that reconstructs the original image, which is the goal of image inversion. This is while FlowOpt results in an optimized latent z_1 , once at hand it could be used to sample z_0 .

The next two algorithm boxes show the concrete differences between FlowChef and FlowOpt.

Algorithm S1 FlowOpt

Require: initialization z_1 , reference image y ,
step size η
for $i \in \{1, 2, \dots, N\}$ **do**
 for $t \in \{1, 1 + \Delta t, \dots, -\Delta t\}$ **do**
 $z_{t+\Delta t} = z_t + v_t(z_t, c)$
 $z_1 \leftarrow z_1 - \eta(z_0 - y)$
Return: initial optimized latent z_1

Algorithm S2 FlowChef

Require: initialization z_1 , reference image y ,
step size η
for $t \in \{1, 1 + \Delta t, \dots, -\Delta t\}$ **do**
 $u_t = v_t(z_t, c)$
 for $i \in \{1, 2, \dots, N\}$ **do**
 $\hat{z}_0 = z_t + u_t$
 $z_t \leftarrow z_t - \eta(\hat{z}_0 - y)$
 $z_{t+\Delta t} = z_t + u_t \Delta t$
Return: denoised sample z_0

Note that in FlowOpt (Alg. S1) the outer loop is over the optimization steps and the inner loop is over the flow timesteps. Namely, in each optimization step, FlowOpt forwards through the entire flow process. In contrast, in FlowChef (Alg. S2) the outer loop is over the flow timesteps and the inner loop is over the optimization steps. Namely, within each outer iteration t , the velocity is evaluated only once for that timestep. This velocity is used to estimate \hat{z}_0 and to update the latent z_t for that timestep such that \hat{z}_0 becomes closer to y .

L LIMITATIONS

Our method, similar to other structure preserving editing methods, has limited performance when large geometrical changes to the image are required. For instance, pose editing, a larger number of optimization steps is required to deviate from the original geometry, and this eventually comes at the cost of diverging from subject’s identity. This can be observed in Fig. S22 when trying to make the dog jump. As can be seen, while the dog begins to perform the jumping action in the later optimization steps, its identity begins to drift away from the original. Moreover, as seen in iteration $i = 5$, the edited dog has 5 legs. This is because our optimization tries to keep the source and target images close in terms of MSE, pushing towards strict alignment, while also trying to adhere to the text, resulting in deviations from the natural image domain. Choosing a more semantic loss function for optimization might remedy these issues, and we leave this for future work.

Another shortcoming of our method is the ability to edit existing text content in images, such as signs, as demonstrated in Fig. S23. We can see that in the first iterations of our optimization process, the method successfully edits the image. Although after several additional iterations, the edit reverts to the original text, failing to adhere to the requested prompt. However, as discussed in Sec. 4, the versatility of our algorithm allows the user to choose between all intermediate editing iterations, mediating this downside. An additional text editing example is presented in Fig. S24, where similar behavior is observed.

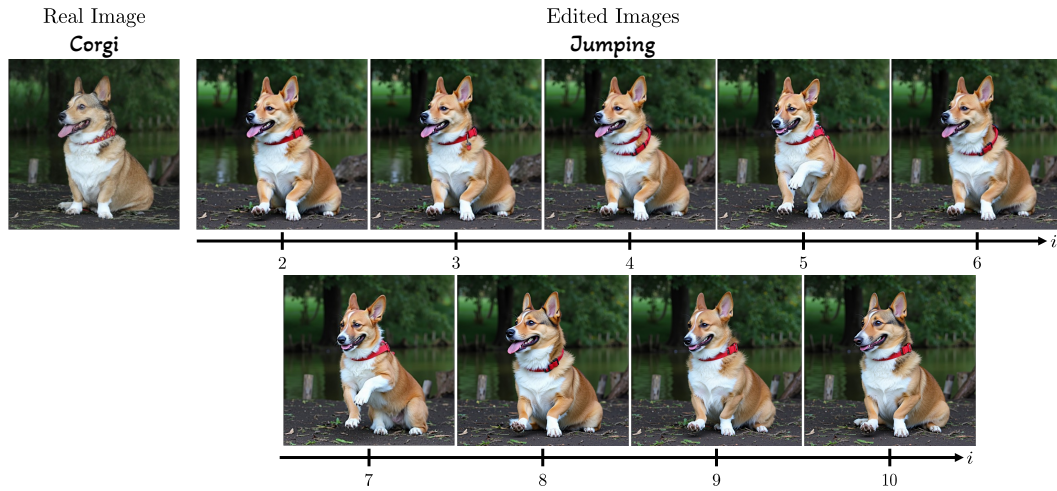


Figure S22: Pose limitation. FlowOpt often fails in preserving the identity of the edited object at the same time of adhering to the target prompt for pose editing.



Figure S23: Text limitation. FlowOpt often fails in text editing as the iterations progress – it would make the result overly similar to the original image.

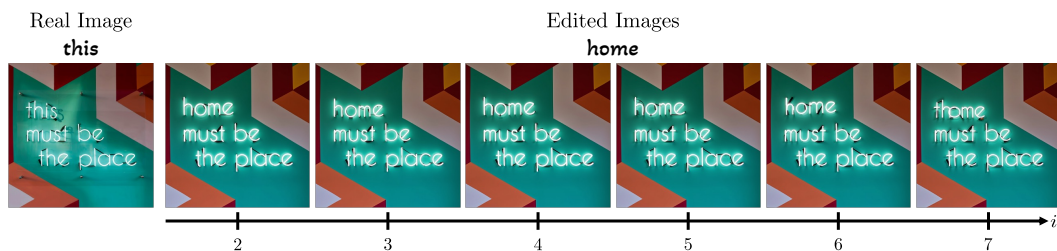


Figure S24: Text limitation. FlowOpt often fails in text editing, as it struggles to preserve the structure of the original image and adhere to the target text prompt simultaneously.