An Edit Friendly DDPM Noise Space: Inversion and Manipulations Supplementary Material

A. Shifting the latent code

As described in Sec. 3, we can shift an input image by shifting its extracted latent code. This requires inserting new columns/rows at the boundary of the noise maps. To guarantee that the inserted columns/rows are drawn from the same distribution as the rest of the noise map, we simply copy a contiguous chunk of columns/rows from a different part of the noise map. In all our experiments, we copied into the boundary the columns/rows indexed $\{50, \ldots, 50 + d - 1\}$ for a shift of d pixels. We found this strategy to work better than randomly drawing the missing columns/rows from a white normal distribution having the same mean and variance as the rest of the noise map. Figure S1 depicts the MSE over the valid pixels that is incurred when shifting the noise maps. This analysis was done using 25 model-generated images. As can be seen, shifting our edit-friendly code results in minor degradation while shifting the native latent code leads to a complete loss of the image structure.

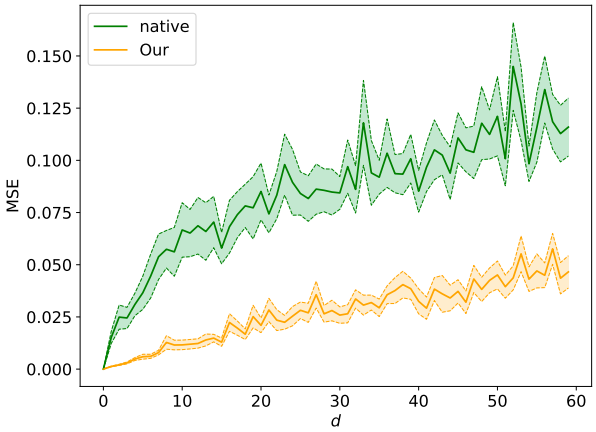

Figure S1. **Shifting the latent code**. We plot the MSE over the valid pixels after shifting the latent code and generating the image. The colored regions represent one standard error of the mean (SEM) in each direction.

B. The effect of the numerical error

In Algorithm 1 we add a correction step for avoiding numerical drifting. This step assists in achieving perfect reconstruction. Note that in order to reconstruct the input image, the hyper-parameters used to extract the noise maps should be identical to the ones used for sampling. Specifically, the prompt, T_{skip}, and strength parameters should be the same in the function $\mu_t(x_t)$ used for the inversion (Eq. 5) and in the function $\mu_t(x_t)$ used during sampling (Eq. 3). We note that the effect of the numerical drifting is noticeable only when using a large strength parameter (see second and third column in Fig. S2). By default, when performing text-based editing, we do not use extreme values for the strength parameter, and therefore in such cases this correction is not needed (rightmost column in Fig. S2).

We calculate the PSNR between the images with and without the correction for the example that appears in Fig. S2. In the reconstruction case, using strength $= 30$, the PSNR can drop to below 17dB. As noted, this correction is not needed for editing, where the PSNR between the edited images with and without the correction is 67.4dB.

Figure S2. **Error correction effect.** Below the images, we specify the strength parameters used for the inversion (first number within the parentheses) and the sampling (second number within the parentheses). Above the images, we specify the prompt used. Above the leftmost column is p_{src}, while above the other columns is p_{tar}. The parameter T_{skip} is set to 36, as in our experiments in the main text. The second and third columns show reconstructions. As can be seen, with a large strength parameter, the reconstruction is not perfect without the correction (*e.g.* the head and the leg of the horse). However, this numerical drifting does not influence the editing quality (rightmost column).

C. CycleDiffusion

As mentioned in Sec. 3.2, CycleDiffusion [6] extracts a sequence of noise maps $\{x_T, z_T, \ldots, z_1\}$ for the DDPM scheme. However, in contrast to our method, their noise maps have statistical properties that resemble those of regular sampling. This is illustrated in Fig. S3, which depicts the per-pixel standard deviations of $\{z_t\}$ and the correlation between z_t and z_{t-1} for CycleDiffusion, for regular sampling, and for our approach. These statistics were calculated over 10 images using an unconditional diffusion model trained on Imagenet, with $\eta = 1.0$, strength $= 3$ and $T_{\text{skip}} = 30$ as hyper parameters. As can be seen, the CycleDiffusion curves are almost identical to those of regular sampling, and are different from ours.

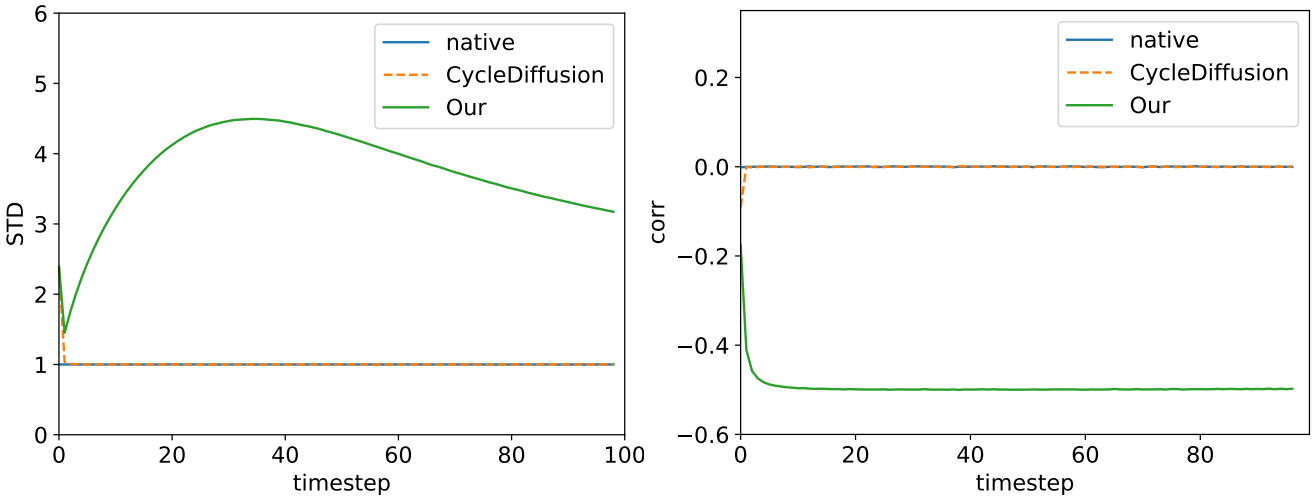

Figure S3. **CycleDiffusion noise statistics.** Here we show the per-pixel standard deviations of $\{z_t\}$ and the per-pixel correlation between them for mssodel-generated images.

The implication of this is that similarly to the native latent space, simple manipulations on CycleDiffusion's noise maps cannot be used to obtain artifact-free effects in pixel space. This is illustrated in Fig. S4 in the context of horizontal flip and horizontal shift by 30 pixels to the right. As opposed to Cycle diffusion, applying those transformations on our latent code, leads to the desired effects, while better preserving structure.

This behavior also affects the text based editing capabilities of CycleDiffusion. In particular, the CLIP similarity and LPIPS distance achieved by CycleDiffusion on the modified ImageNet-R-TI2I dataset are plotted in Fig. 10. As can be seen, when tuned to achieve a high CLIP-similarity (*i.e.* to better conform with the text), CycleDiffusion's LPIPS loss increases significantly, indicating that the output images become less similar to the input images. For the same level of CLIP similarity, our approach achieves a substantially lower LPIPS distance.

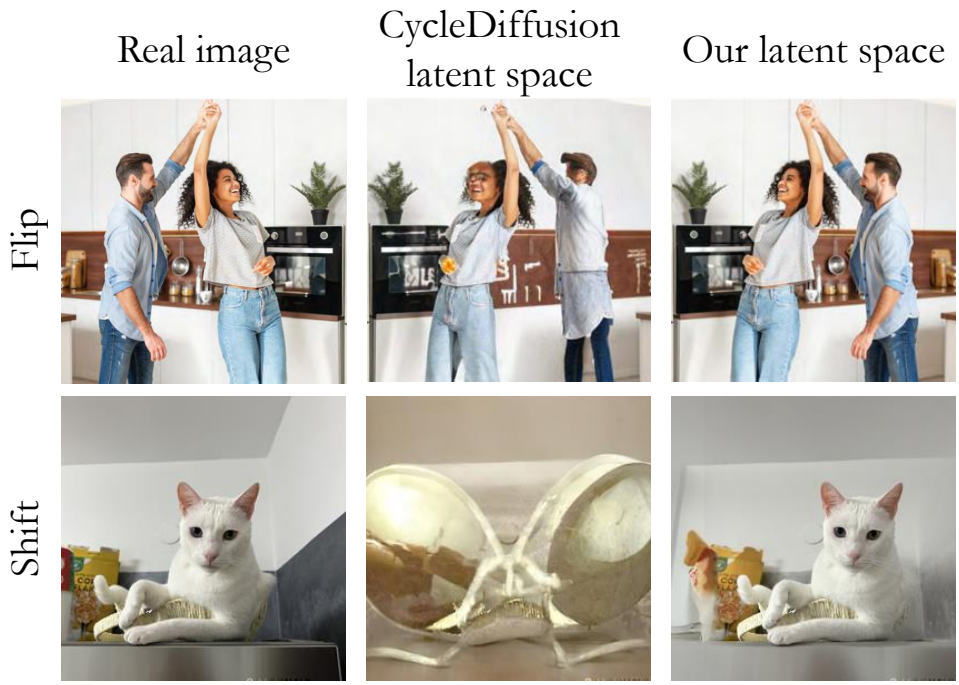

Figure S4. **Flip and shift with CycleDiffusion and with our inversion.**

D. The effect of skip and strength parameters

Recall from Sec. 4 that in order to perform text-guided image editing using our inversion, we start by extracting the latent noise maps while injecting the source text into the model, and then generate an image by fixing the noise maps and injecting a target text prompt. Two important parameters in this process are T_{skip}, which controls the timestep $(T - T_{\text{skip}})$ from which we start the generation process, and the strength parameter of the classifier-free scale [2]. Figure S5 shows the effects of these parameters. When T_{skip} is large, we start the process with a less noisy image and thus the output image remains close to the input image. On the other hand, the strength parameter controls the compliance of the output image with the target prompt.

Figure S5. **The effects of the skip and the strength parameters**.

E. Integrating to P2P

As described in sec. 4, our inversion method can be integrated with existing editing methods that rely on DDIM inversion. In addition to combining our method with Zero-Shot I2I, we assess the integration with Prompt-to-Prompt (P2P) [1]. In that case, we decrease the hyper-parameter controlling the cross-attention from 0.8 to 0.6 (as our latent space already strongly encodes structure). We note that P2P has different modes for different tasks (swap word, prompt refinement), and we chose its best mode for each image-prompt pair. Figure2 S6 and S7 show that P2P does not preserve structure well. Yet, P2P does produce appealing results when used with our inversion.

Figure S6. **Comparison to P2P, with and without our inversion**.

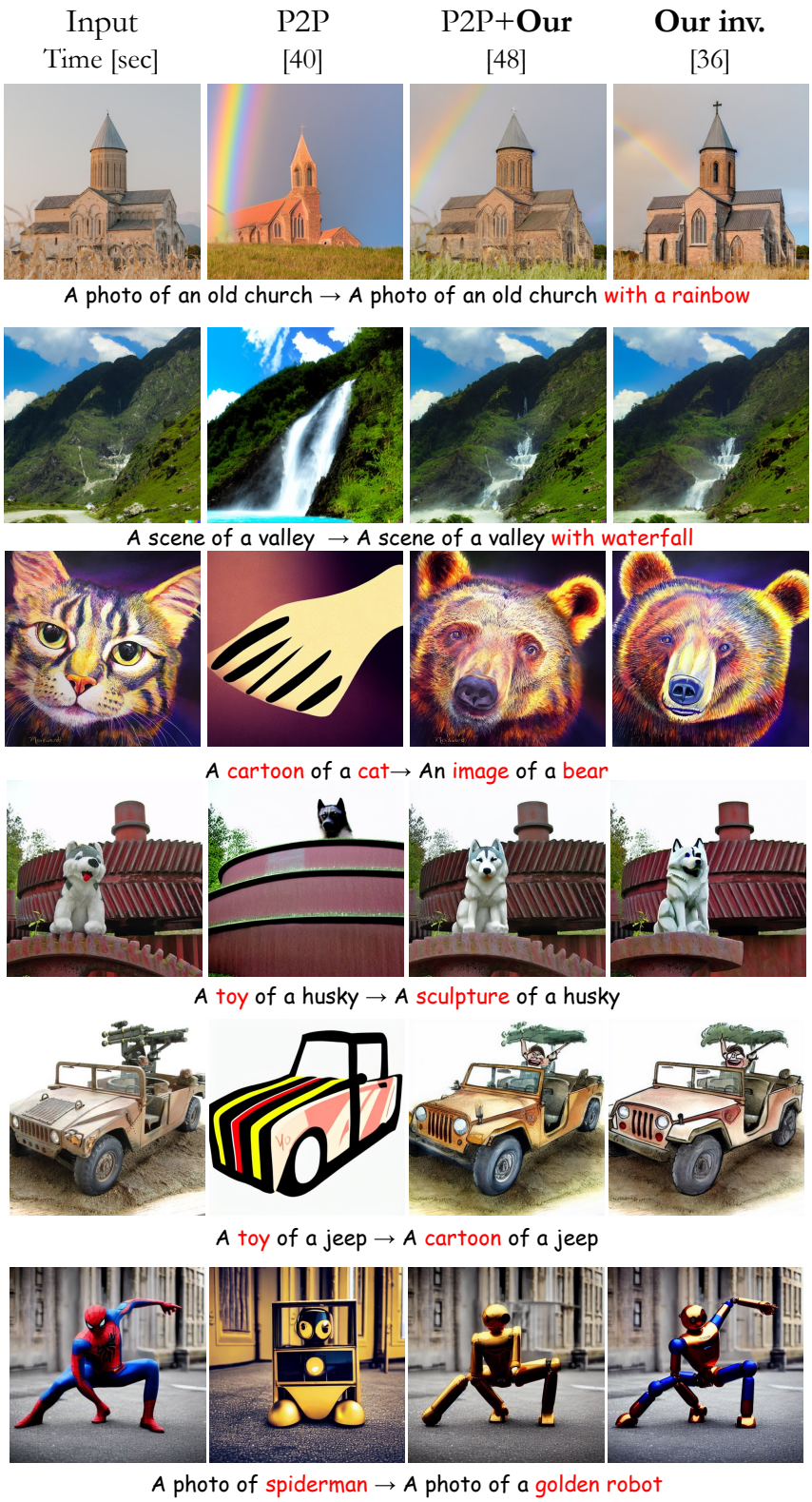

Figure S7. **Additional comparisons to P2P, with and without our inversion**.

F. Additional details on experiments and further numerical evaluation

For all our text-based editing experiments, we used Stable Diffusion as our pre-trained text-to-image model. We specifically used the StableDiffusion-v-1-4 checkpoint. We ran all experiments on an RTX A6000 GPU. We now provide additional details about the evaluations reported in the main text. All datasets and prompts will be published.

In addition to measuring CLIP-based scores, LPIPS scores, and running time, we also measure the diversity among generated outputs (higher is better). Specifically, for each image and source text p_{src}, we generate 8 outputs with target text p_{tar} and calculate the average LPIPS distance over all $\binom{8}{2}$ pairs.

F.1. Experiments on the modified ImageNet-R-TI2I

Our modified ImageNet-R-TI2I dataset contains 44 images: 30 taken from PnP [5], and 14 from the Internet and from the code bases of other existing text-based editing methods. We verified that there is a reasonable source and target prompt for each image we added. For P2P [1] (with and without our inversion), we used the first 30 images with the "replace" option, since they were created with rendering and class changes. That is, the text prompts were of the form "a ≪rendering≫ of a ≪class≫" (*e.g.* "a sketch of a cat" to "a sculpture of a cat"). The last 14 images include prompts with additional tokens and different prompt lengths (*e.g.* changing "A photo of an old church" to "A photo of an old church with a rainbow"). Therefore for those images we used the "refine" option in P2P. We configured all methods to use 100 forward and backward steps, except for PnP whose supplied code does not work when changing this parameter.

Table S1 summarizes the hyper-parameters we used for all methods. These apply to both the numerical evaluations and to the visual results shown in the figures. For our inversion, for P2P with our inversion, and for CycleDiffusion we arrived at those parameters by experimenting with various sets of parameters and choosing the configuration that led to the best CLIP loss under the constraint that the LPIPS distance does not exceed 0.3. For DDIM inversion and for P2P (who did not illustrate their method on real images), such a requirement could not be satisfied. Therefore for those methods we chose the configuration that led to the best CLIP loss under the constraint that the LPIPS distance does not exceed 0.62. We show results over DDIM inversion with 100 and 50 number of diffusion steps. For DDIM inversion mid-way we use the inversion until a specific timestep. For PnP, null-text inversion, and EDICT we used the default parameters supplied by the authors.

Method	#inv. steps	#edit steps	strength	T_{skip}	$\tau_{\text{x}}/\tau_{\text{a}}$
DDIM inv. ($T = 100$)	100	100	9	0	–
DDIM inv. ($T = 50$)	50	50	9	0	–
P2P	100	100	9	0	80/40
P2P + Our inv.	100	100	9	12	60/20
PnP	1000	50	10	0	40/25
EDICT	50	50	3	10	—
null-text inversion	50	50	7.5	0	80/40
CycleDiffusion ($\eta = 0.1$)	100	100	3	30	–
CycleDiffusion ($\eta = 1.0$)	100	100	3	30	–
Our inv.	100	100	15	36	–

Table S1. **Hyper-parameters used in experiments on the modified ImageNet-R-TI2I dataset**. The parameter 'strength' refers to the classifier-free scale of the generation process. As for the strength used in the inversion stage, we set it to 3.5 for all methods except for PnP and CycleDiffusion which uses 1. The timestep at which we start the generation is $T - T_{\text{skip}}$ and, in case of injecting attentions, we also report the timesteps determine until which step (starting from zero) the cross- and self-attentions are injected, τ_{x} and τ_{a} respectively.

Table S2 and Figure S8 summarizes the comparisons of all methods reported in the paper with the hyper-parameters from Tab S1. The results show that our inversion achieves a good balance between LPIPS and CLIP, while requiring short edit times. Integrating our inversion into P2P improves their performance in both metrics. Our method, CycleDiffusion, and null-text inversion support diversity among generated outputs.

Method	CLIP sim.↑	LPIPS↓	Diversity↑	Time↓
DDIM inv. ($T = 100$)	0.31	0.62	0.00	39
DDIM inv. ($T = 50$)	0.31	0.62	0.00	39
P2P	0.30	0.61	0.00	40
P2P+Our	0.31	**0.25**	0.11	48
PnP	0.31	0.36	0.00	206
EDICT	0.29	0.27	0.00	520
null-text inversion	0.29	0.35	0.08	160
CycleDiffusion, $\eta = 0.1$	0.30	0.27	0.21	**36**
CycleDiffusion, $\eta = 1.0$	0.30	0.26	**0.306**	**36**
Our inv.	**0.32**	0.29	0.18	**36**

Table S2. Evaluation on modified ImageNet-R-TI2I dataset.

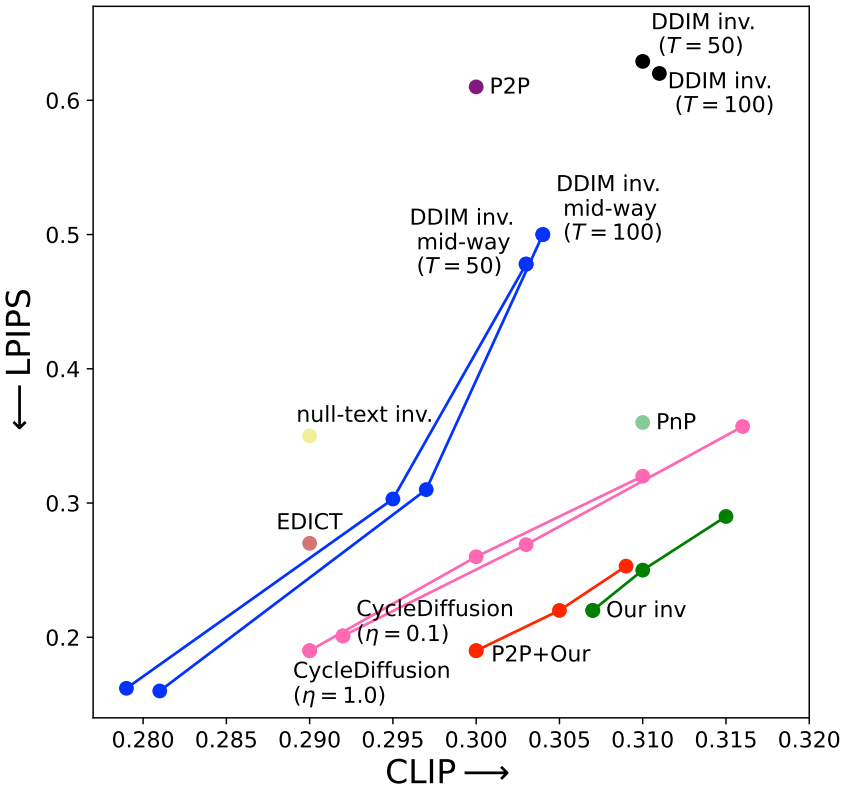

Figure S8. **Fidelity to source image vs. compliance with target text.** We show a comparison of all methods.

F.2. Experiments on the modified zero-shot I2IT dataset

The second dataset we used is the modified Zero-Shot I2IT dataset, which contains 4 categories (cat, dog, horse, zebra). Ten images from each category were taken from Parmar *et al*. [4], and we added 5 more images from the Internet to each category. Zero-Shot I2I [4] does not use source-target pair prompts, but rather pre-defined source-target classes (*e.g*. cat↔dog). For their optimized DDIM-inversion part, they use a source prompt automatically generated with BLIP [3]. When combining our inversion with their generative method, we use $T_{\text{skip}} = 0$ and an empty source prompt. Table S3 summarizes the hyper-parameters used in every method.

Method	#inv. steps	#edit steps	strength	T_{skip}	λ_{xa}
Zero-Shot	50	50	7.5	0	0.1
Zero-Shot+Our	50	50	7.5	0	0.03

Table S3. **Hyper-parameters used in experiments on the modified Zero-Shot I2IT dataset**. In this method, cross-attention guidance weight is the parameter used to control the consistency in the cross-attention maps, denoted here as λ_{xa}. We set the strength (classifier-free scale) in the inversion part to be 1 and 3.5 for "Zero-shot" and "Zero-shot+Our" respectively.

Tab S4 summarizes the comparison to the Zero-shot method. The results show that integrating our inversion improves the similarity to the input image while keeping the CLIP accuracy high. We also exhibit non-negligible diversity among the generated outputs

Method	CLIP Acc.↑	LPIPS↓	Diversity↑	Time
Zero-Shot	0.88	0.35	0.07	**45**
Zero-Shot+Our	**0.88**	**0.27**	**0.16**	46

Table S4. Evaluation on the modified Zero-Shot I2IT dataset.

G. Additional results

Due to the stochastic nature of our method, we can generate diverse outputs, a feature that is not naturally available with methods relying on the DDIM inversion. Figures S9 and S10 show several diverse text-based editing results. Figures S11 and S12 provide further qualitative comparisons between all methods tested on the ImageNet-R-TI2I dataset.

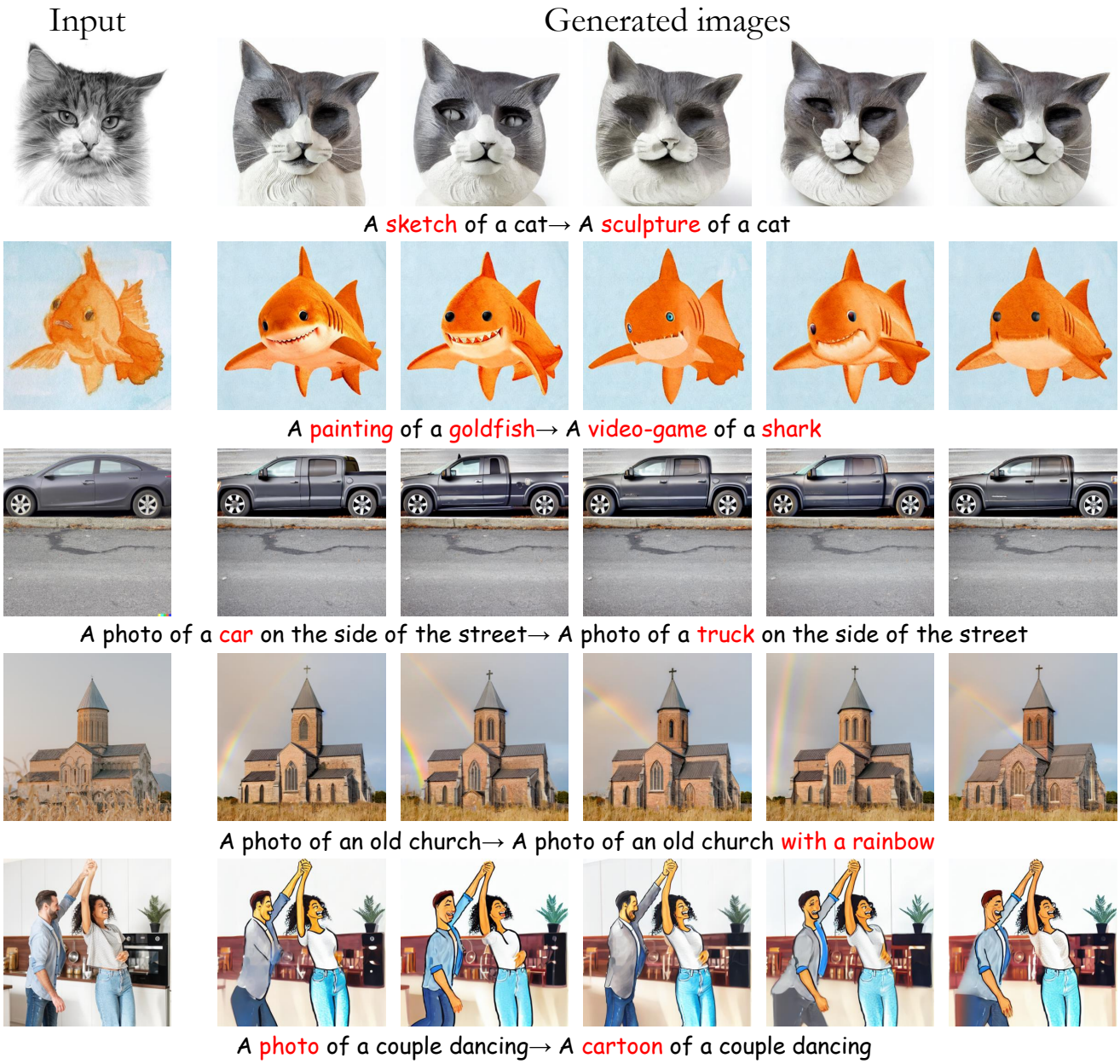

Figure S9. **Diverse text-based editing with our method**. We apply our inversion five times with the same source and target prompts (shown beneath each example). Note how the variability between the results is not negligible, while all of them conform to the structure of the input image and comply with the target text prompt. Notice *e.g.* the variability in the sculpture cat's eyes and mouth, and how the rainbow appears in different locations and angles.

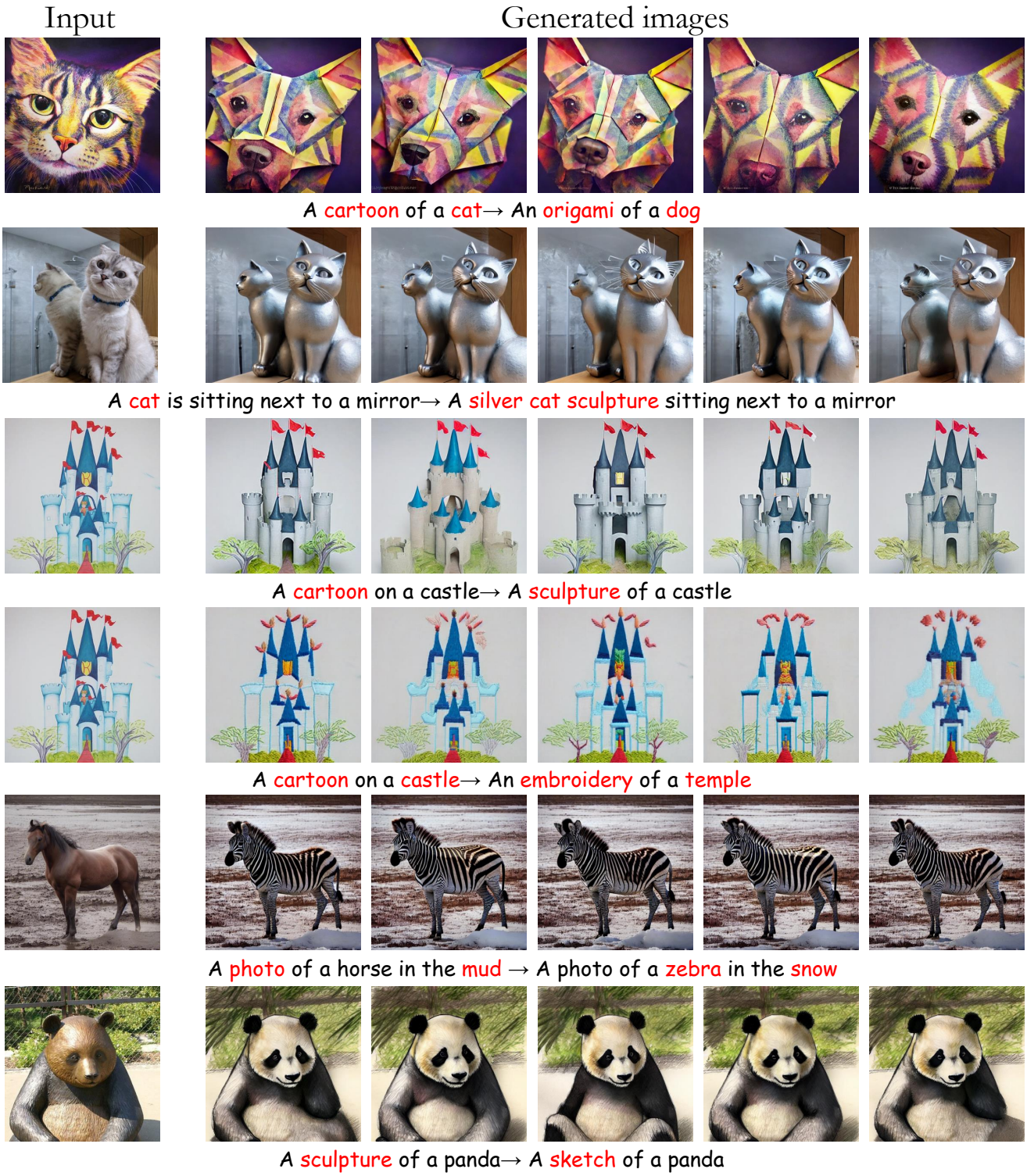

Figure S10. **Additional results for diverse text-based editing with our method**. Notice that each edited result is slightly different. For example, the eyes and nose of the origami dog change between samples, and so do the zebra's stripes.

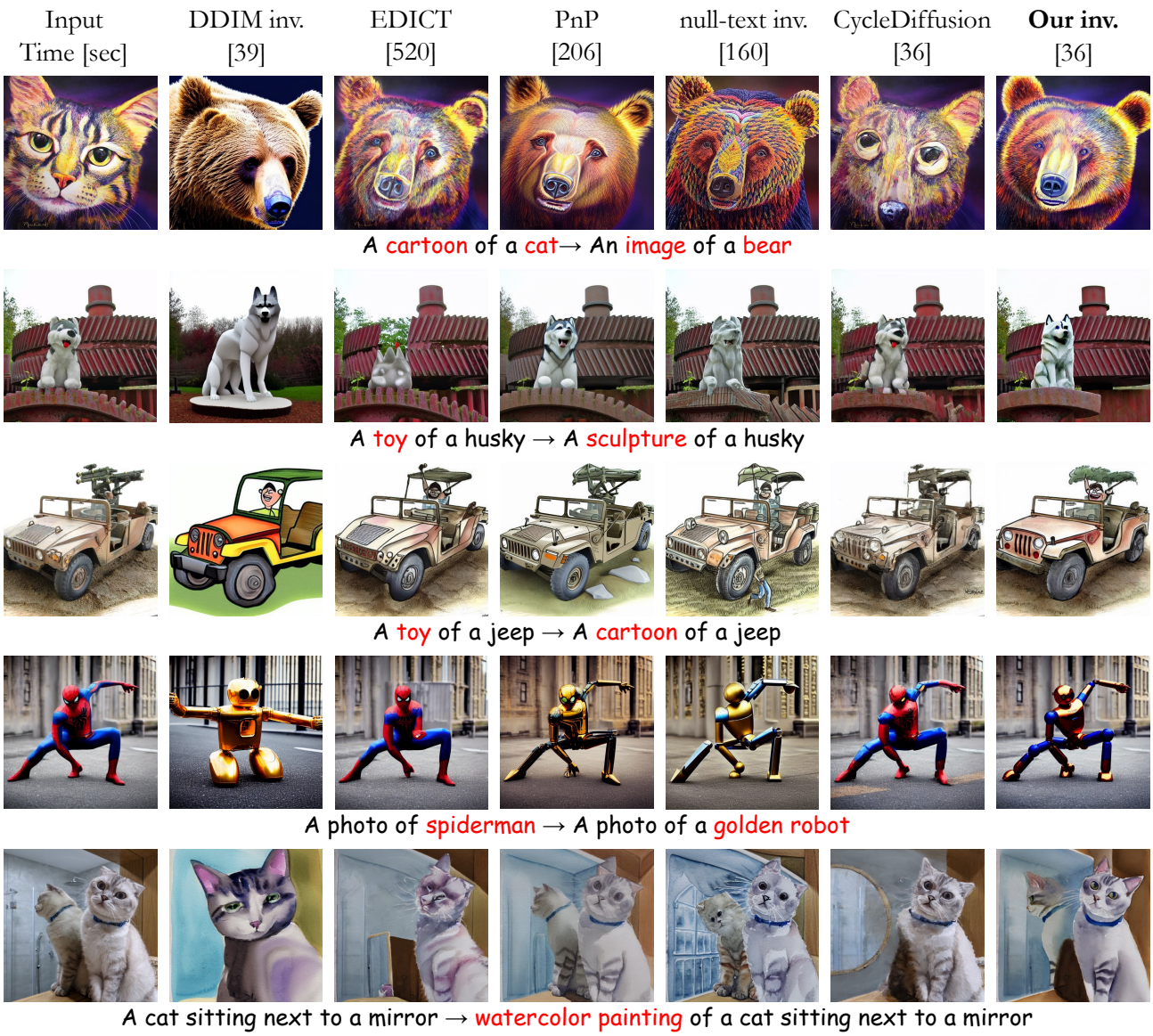

Figure S11. **Qualitative comparisons between all methods.**

Figure S12. **Additional qualitative comparisons between all methods.**

References

[1] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations (ICLR)*, 2023. 6, 9

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

[3] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, pages 12888–12900, 2022. 11

[4] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 11

[5] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2023. 9

[6] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3