Supplementary Materials for LoMOE: Localized Multi-Object Editing via Multi-Diffusion

Contents

1 INTRODUCTION

To keep the overall manuscript self-contained, we include additional details in the supplementary material. The source code for LoMOE along with the LoMOE-Bench dataset will be released in due time.

2 METHOD DETAILS

Specific aspects of the framework, including regularized inversion and temperature scaling, are described below.

2.1 Regularised Inversion

To softly enforce gaussianity on the inverted noise maps generated during the DDIM Inversion, we use a pairwise regularization $\mathcal{L}_{pair}$ [19] and a divergence loss $\mathcal{L}_{KL}$ [12] weighted by λ (refer Sec. 3.1 of the main paper). These losses ensure that there is **(1)** no correlation between any pair of random locations and **(2)** zero mean, unit variance at each spatial location, respectively. Mathematically, the pairwise regularization loss is given by:

$$\mathcal{L}_{\text{pair}} = \sum_{p} \frac{1}{S_p^2} \sum_{\delta=1}^{S_p-1} \sum_{x,y,c} \eta^p_{x,y,c} \left(\eta^p_{x-\delta,y,c} + \eta^p_{x,y-\delta,c} \right) \qquad (1)$$

where $\{\eta^0, \eta^1, \cdots, \eta^p\}$ denote the noise maps with size S_p at the p^{th} pyramid level, δ denotes the offset which helps propagate long-range information [11, 19], and $\{x, y, c\}$ denotes a spatial location. Here, we set $p = 4$ and $\eta^0 = \epsilon_\theta \in \mathbb{R}^{64\times64\times4}$, where the subsequent noise maps are obtained via max-pooling.
The divergence loss is given by:

$$\mathcal{L}_{KL} = \sigma^2_{\epsilon_\theta} + \mu^2_{\epsilon_\theta} - 1 - \log(\sigma^2_{\epsilon_\theta} + \varepsilon) \qquad (2)$$

where μ_{ϵ_θ} and $\sigma^2_{\epsilon_\theta}$ denotes the mean and variance of ϵ_θ and ε is a stabilization constant.

2.2 Temperature Scaling

Given a vector $z = (z_1, \cdots, z_n) \in \mathbb{R}^n$, it can be transformed into a probability vector via

$$\text{Softmax}(z|\tau)_i = \frac{e^{z_i/\tau}}{\sum_{j=1}^{n} e^{z_j/\tau}} \qquad (3)$$

where τ is a temperature parameter [8] which varies the smoothness of the output distribution. In general, lower values of τ result in a sharp distribution, and increasing τ softens the distribution. This method has been used in applications such as model calibration [9], image restoration [20] and image inpainting [23]. In this work, we use a constant temperature scale to ensure the distributional smoothness of the cross-attention maps, setting $\tau = 1.25$. Further ablation on τ is discussed in Sec. 3.

3 ABLATION STUDY

In addition to the quantitative ablation of λ_{xa} and λ_b, we further study the impact of varying the temperature scaling parameter τ and bootstrap T_b. Specifically, we experiment for $\tau \in \{1.00, 1.25, 1.50, 1.75, 2.0\}$ and $T_b \in \{5, 10, 20, 30, 35\}$ and report the results in Table 1. We also do a visual ablation for the effect of the tightness of the mask on the performance of LoMOE. We also explore varying the tightness of the mask. We find that as long as the object to be edited is fully masked, the model performs well, as depicted in Figure 3.

3.1 Temperature Scaling

The results for variation in τ are summarized in Tab. 1 and have been depicted visually in Fig. 1. We observe that the edited image tends to go towards the source image with an increase in τ, which can be attributed to over-smoothing the distribution. This is also indicated by the *neural* metrics in Tab. 1, where an increase in τ results in increasing source CS and a decreasing target CS. This is further exemplified by the *background* metrics and Structural Distance, which are the best for $\tau = 2.00$. In this work, we set $\tau = 1.25$ as mentioned in Sec. 3.4 of the main paper. This choice of τ resulted in visually pleasing edits and we observed semantically coherent outputs for $\tau \in [1, 1.5]$.

3.2 Bootstrap

Upon analyzing the findings presented in Tab. 1, we opt for $T_b = 10$ based on the observation that the general structure and overall layout of the image is established within the first 10 denoising steps. Subsequently, the diffusion model manifests the finer details of the image, in accordance with [2]. We also observe using a higher value of bootstrap aids in *addition*-based edits.

τ	T_b	Source CLIP Score (↑)	Background LPIPS (↓)	Background PSNR (↑)	Background SSIM (↑)	Structural Distance (↓)	Target CLIP Score (↑)
1.00	-	23.4216	0.0586	30.1023	0.8822	0.0728	25.9163
1.25	-	**23.7507**	0.0522	30.4707	0.8849	0.0715	**26.0902**
1.50	-	24.1785	0.0497	30.7565	0.8863	0.0708	25.7919
1.75	-	25.0428	0.0466	31.1206	0.8875	0.0709	24.9769
2.00	-	25.4275	**0.0409**	**31.5829**	**0.8896**	**0.0652**	24.1544
-	05	23.5422	0.0562	30.1123	0.8838	0.0782	25.9403
-	10	**23.5445**	**0.0546**	**30.3154**	**0.8847**	**0.0710**	**26.0740**
-	20	23.4344	0.0587	30.0937	0.8822	0.0723	25.8746
-	30	23.4494	0.0618	29.8495	0.8792	0.0757	25.9404
-	35	23.2644	0.0621	29.8123	0.8792	0.0774	25.8089

Table 1: Further Ablation: We experiment with different values of the temperature parameter (τ) and bootstrap (T_b) parameters. From the *neural* and *background* metrics, we observe that the similarity between the edited and the input image increases for higher values of τ and that $T_b = 10$ is the optimal value for the bootstrap parameter.

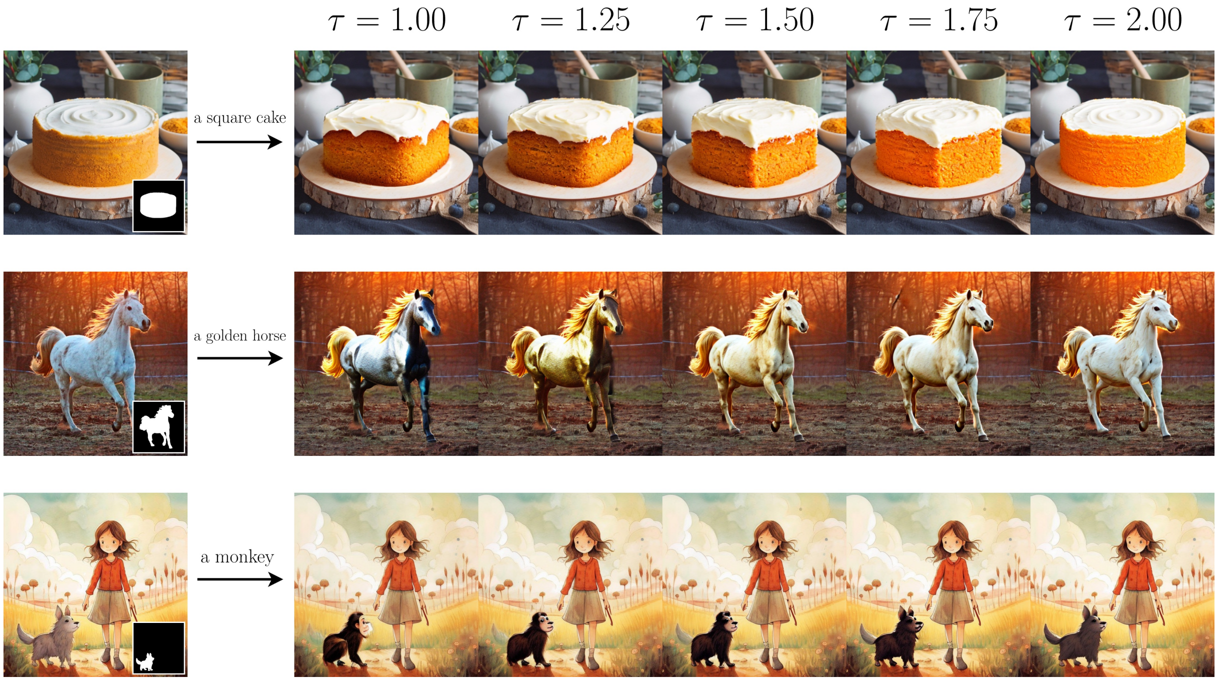


Figure 1: Ablation on Temperature Scaling: Impact of an increasing temperature parameter, τ's on the edits. We observe that an increase in the value of τ results in the edited image moving towards the input image. Empirically, we see visually appealing edits are achieved at $\tau = 1.25$.

3.3 Inversion

As mentioned in Sec. 3.1 of the main paper, *inversion* helps initiate the editing procedure and ensures a coherent and controlled edit. To understand the impact of *inversion*, we compare two different initializations for the *edit* process (refer Sec. 3.2 of the main paper), namely **(1)** $x_T = x_{inv}$ and **(2)** $x_T = \zeta$. Here, $\zeta \in \mathbb{R}^{64\times64\times4}$ denotes a random latent with elements sampled from $\mathcal{N}(0,1)$. Specifically, we choose to showcase this impact on *style transfer* based edits.

From Fig. 2, we observe that the images *with inversion* are structurally much closer to the input image compared to the ones generated using a *random latent*, which is also indicated by the Structural Distance metric. In most cases, although using a random latent generates a faithful edit to the given prompt, it changes the content of the image, resulting in undesirable outputs. Therefore, using inversion is crucial for faithful image editing.

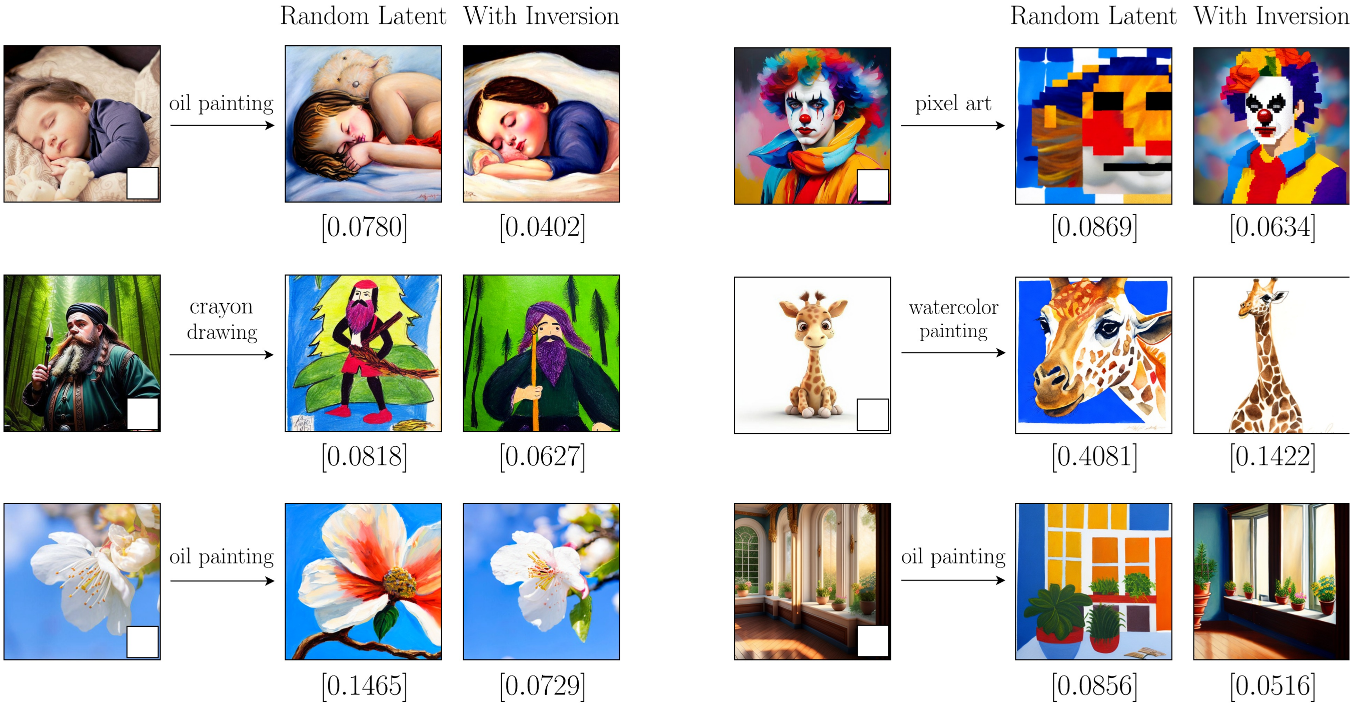


Figure 2: Ablation on Inversion: We study the impact of editing with a *random latent* compared to initiating the editing process via *inversion*. The outputs from LoMOE for both cases are captioned with the [Structural Distance (↓)]. We observe that the structural similarity is preserved when using *inversion* instead of a *random latent* to initiate the editing process.

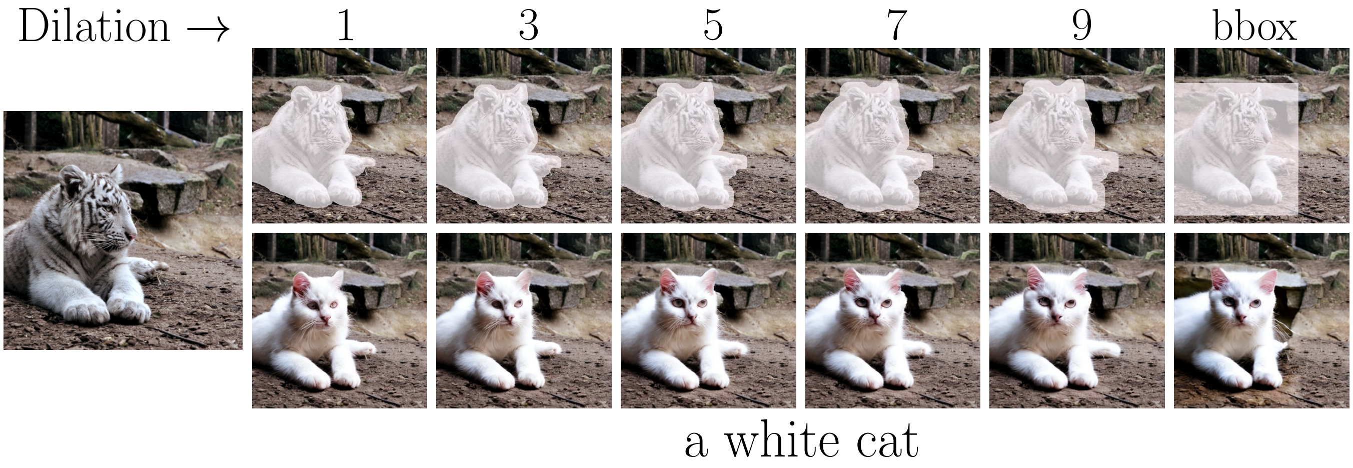


Figure 3: Illustrating the impact of enlarging the mask beyond the target object in LoMOE. As the mask undergoes progressive dilation, culminating in the use of a bounding box as the mask, it becomes evident that such transformations have minimal effect.

3.4 Inference Time

In a multi-object scenario, LoMOE separates itself by executing all edits in a single pass, resulting in substantial time savings compared to iterative methods. This is highlighted in Table 2, where our approach proves particularly advantageous in scenarios involving multiple objects, demonstrating a notable decrease in edit time. Unlike other methods that run iteratively to generate multi-object edits, LoMOE's streamlined approach minimizes the need for repeated computations, enhancing overall efficiency. The gains in edit time underscore LoMOE's practical applicability in real-world editing tasks, showcasing its potential to streamline and expedite complex multi-object editing processes. The peak memory utilization (PMU) using the setup in Sec. 3.4 of the main paper has been detailed in Table 3. Additionally, the time taken to optimize each objective is as follows (Ref. Main Paper Sec. 3):

$$\text{start} \rightarrow \mathcal{L}_{xa}(\text{Eqn. (9)}) : 0.165025 \text{ sec}$$
$$\mathcal{L}_{xa}(\text{Eqn. (9)}) \rightarrow \mathcal{L}_{b}(\text{Eqn. (10)}) : 0.001386 \text{ sec}$$
$$\mathcal{L}_{b}(\text{Eqn. (10)}) \rightarrow y^{*}_{t-1}(\text{Eqn. (13)}) : 0.185235 \text{ sec}$$

Method	Inference Time for N masks (sec)				
	1	2	3	5	7
GLIDE	22.10	41.10	63.76	106.99	153.11
DiffEdit	22.25	41.30	65.91	110.85	152.60
BLD	27.20	51.60	80.40	135.24	185.37
SDInpaint	29.43	49.02	71.91	116.34	158.40
Iterative	24.97	45.75	70.50	117.35	162.37
LoMOE	23.19 **(7.1)**	31.3 **(31.6)**	39.35 **(44.2)**	55.47 **(52.7)**	76.15 **(53.1)**

Table 2: In a multi-object setting, we report the inference time of all the methods for varying number of masks N. *Iterative* denotes the average runtime of GLIDE, DiffEdit and BLD. We report the percentage improvement by LoMOE over *Iterative* (in green)

Method	Peak Memory Utilization for N masks (MiB)				
	1	2	3	5	7
inversion	———-11029———-				
reconstruction	———-6813———-				
edit	12609	17127	21367	30235	38979

Table 3: Peak Memory Utilization (PMU) by LoMOE.

4 EXPERIMENTAL PROTOCOL

4.1 Datasets

To facilitate a comparison between various baselines on *single-object* edits, we employ a modified subset of the PIE-Bench [10] dataset supplemented with images from AFHQ [5], COCO [14], and Imagen [22]. Overall, the benchmark consists of 300 images, covering editing types such as changing objects, adding objects, changing object content, changing object color, changing object material, changing background, and changing image style. Sample images for each edit type are shown in Fig. 14.

The newly proposed *multi-object* editing benchmark LoMOE-Bench consists of 64 images, covering various editing types, with each image featuring 2 to 7 masks paired with corresponding text prompts. The masks for the images in LoMOE-Bench and the supplemental images in the *single-object* dataset are generated using SAM [13]. In practice, the user is required to provide a bounding box around the object via a GUI interface, which then automatically saves the segmented mask, as mentioned on Sec. 4.1 of the main paper. Sample images from LoMOE-Bench are depicted in Fig. 18. We further observe the variety of objects in LoMOE-Bench images and their masks in Fig. 4 and Fig. 5 shows the name of the objects being edited in the form of a word cloud. The images are also supplemented with various text-based annotations used by different baselines (refer Table 4) via a JSON file, including

Method	Image	Mask	TIP	SMP	TMP	EIn
SDEdit [16]	✓	✗	✓	✗	✗	✗
I-P2P [3]	✓	✗	✗	✗	✗	✓
NTI (w/ P2P) [17]	✓	✗	✓	✗	✗	✗
MasaCtrl [4]	✓	✗	✓	✗	✗	✗
DiffEdit [6]	✓	✓	✗	✓	✓	✗
GLIDE [18]	✓	✓	✗	✗	✓	✗
BLD [1]	✓	✓	✗	✗	✓	✗
SDInpaint [21]	✓	✓	✗	✗	✓	✗
LoMOE	✓	✓	✗	✗	✓	✗

Table 4: Annotations required by various baseline methods included in the modified *single-object* dataset and LoMOE-Bench.

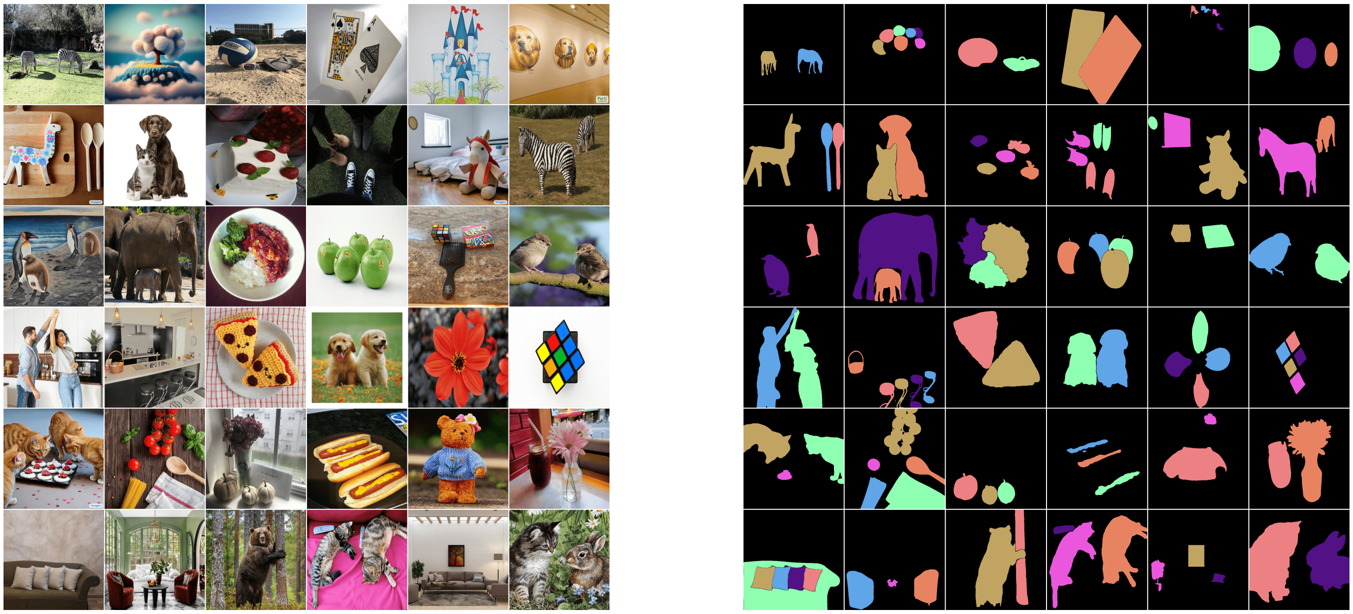
- **Target Image Prompt (TIP)**: A complex prompt describing the complete image after the edit.
- **Source Mask Prompt (SMP)**: A simple text prompt describing the object inside the masked region of the input image.
- **Target Mask Prompt (TMP)**: A simple text prompt that describes the edited object inside the masked region.
- **Edit Instruction (EIn)**: Edit instruction for I-P2P [3].

The dataset statistics are the following. On average, masks cover 8.34% of the image (standard deviation: 10.1%), with each image containing an average of 3 masks (standard deviation: 1.17). In addition, we see the spatial distribution of the masks in a heat map in Fig. 6, demonstrating the frequency with which each pixel is masked.

4.2 Baselines

We use the official implementation for all baseline methods using PyTorch, except for DiffEdit as the code has not been made public. SDEdit uses the target prompt for text-guided image editing and does not require any other input. DiffEdit by construction uses the DDIM solver, but the unofficial implementation uses DPM solver [15] for better sample efficiency. The method also generates noisy masks based on the source and target mask prompts, thus we choose to use the masks in the dataset (as mentioned in Sec. 4.2 of the main paper).

I-P2P requires an edit instruction along with the image and does not need any other inputs. For example, the edit instruction for the first image in Fig. 13 would look like: *“change the shape of the cake to a square"*. It is also important to note that although all other methods use the pre-trained Stable Diffusion model directly, Instruct-P2P is trained by finetuning this model. Finally, GLIDE and BLD are similar to LoMOE in that they only require the target mask prompt as additional inputs.



(a) Images to be edited

(b) Masks corresponding to the real images

Figure 4: LoMOE-Bench Dataset: Dataset comprising images with multiple edit targets along with their corresponding masks (a) and (b), and a word cloud

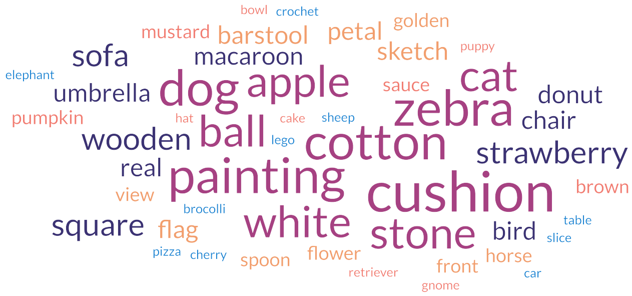


Figure 5: Word Cloud depicting the variety of objects present in LoMOE-Bench

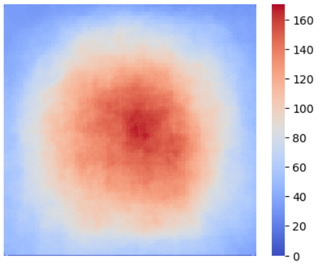


Figure 6: Heatmap showing the number of masks that cover a certain pixel in 512×512 images.

4.3 Additional Results

We supplement the qualitative results provided in the main paper (Ref. Main Paper Sec. 5) by comparing LoMOE against baselines on more single-object edits, depicted in Fig. 13. Furthermore, we showcase single-object and multi-object edits with LoMOE in Figs. 14 and 18 for samples from LoMOE-Bench and the *single-object* benchmark for multiple masks and various edit types, respectively.

5 LIMITATIONS

In Figures 8, 9 and 7, we analyze the effect of lighting, shadows and reflections in the outputs produced by LoMOE. Additionally, Figures 10 and 11 delve into object coherence and some failure cases of LoMOE.

Lighting: In Fig. 8, we observe that the edited image has similar lighting properties to those of the source image.

Reflections: In Fig. 9, we observe that LoMOE adds in realistic reflections for *addition* based examples, based on the mask used. For example, there is no reflection in the "blue fish“ example due to the mask constraint. This opens up avenues for a mask-free version of LoMOE. Additionally, in the example of a "sunny day“, the LoMOE uses the reflections in the water as a prior to aptly fill in the masked region.

Shadows: In Fig. 7, we observe that LoMOE attempts to add realistic shadows.

Realism: In Fig. 10, we demonstrate object-background coherence using a *background* change and an *addition* example. In both cases, LoMOE produces realistic and coherent results.

Failures: In Fig. 11, in cases where reflections fall outside the mask, LoMOE cannot remove them due to the mask-constraint, resulting in inconsistent lighting. Furthermore, LoMOE can perform global edits like style-transfer (Ref. Fig. 14) but cannot facilitate spatial manipulation like swapping or translation while preserving the identity and style of the object. While translation can be achieved through deletion and addition, as demonstrated in Fig. 11, it doesn't maintain the object's identity. Textual inversion [7] might be a potential solution to preserve identity for future works.

6 BROADER IMPACT

Generative image editing models are powerful tools that can create realistic and diverse images from text or other inputs. They have many potential applications in domains such as art, entertainment, education, medicine, and security. However, they also pose significant ethical and social challenges that need to be addressed. Some of these challenges include:

- The risk of generating harmful or offensive images that may violate human dignity, privacy, or rights.
- The possibility of manipulating or deceiving people with fake or altered images that may affect their beliefs, emotions, or behaviours.
- The difficulty of verifying the authenticity or provenance of images that may have legal or moral implications.
- The impact of replacing or reducing human creativity and agency with automated or algorithmic processes.

These challenges require careful consideration and regulation from various perspectives, such as technical, legal, ethical, and social. However, we believe that despite these drawbacks, better content creation methods will produce a net positive for society. Furthermore, we advocate for conducting such research in the public domain, emphasizing transparency and collaborative efforts to ensure responsible and beneficial outcomes for the broader community.

REFERENCES

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. 2023. Blended Latent Diffusion. *ACM Trans. Graph.* 42, 4, Article 149 (jul 2023), 11 pages. https://doi.org/10.1145/3592450

[2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. 2023. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. In *ICML*. PMLR.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18392–18402.

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. 2023. 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)*. 22560–22570.

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[6] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2023. DiffEdit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=3lge0p5o-M-

[7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2019. On calibration of modern neural networks. (2019).

[9] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. 2018. Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 41–50. https://api.semanticscholar.org/CorpusID:55700923

[10] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. 2024. PnP Inversion: Boosting Diffusion-based Editing with 3 Lines of Code. *International Conference on Learning Representations (ICLR)* (2024).

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 8107–8116. https://doi.org/10.1109/CVPR42600.2020.00813

[12] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations, ICLR 2014*, Yoshua Bengio and Yann LeCun (Eds.).

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *arXiv:2304.02643* (2023).

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[15] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *arXiv preprint arXiv:2206.00927* (2022).

[16] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073* (2021).

[17] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6038–6047.

[18] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning*. PMLR, 16784–16804.

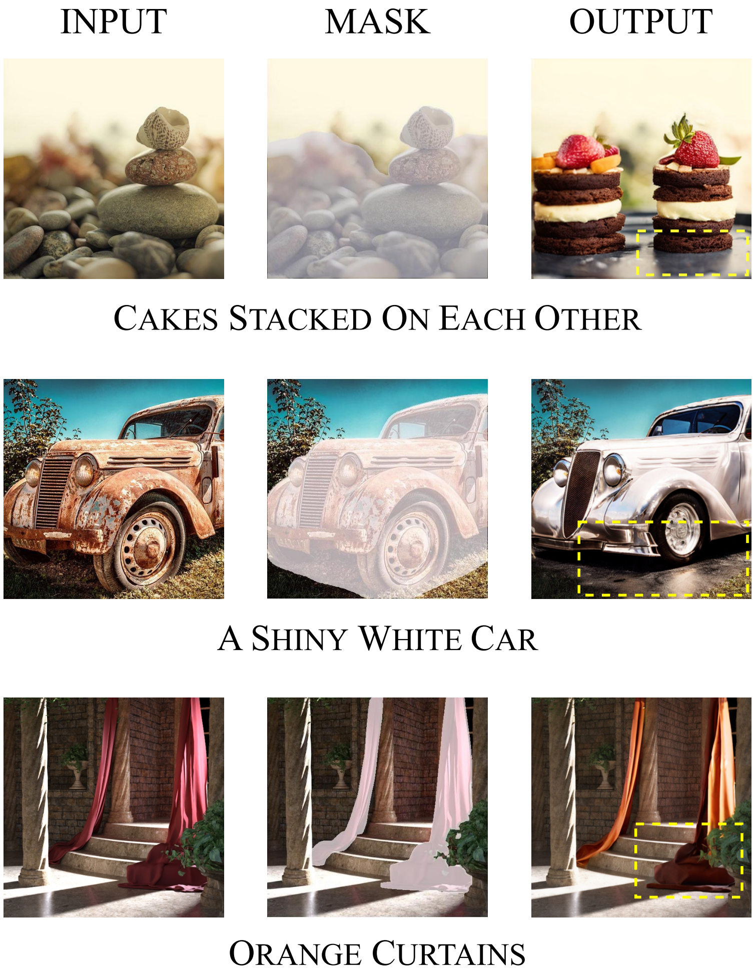
[19] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. 2023. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–11.

[20] Tobias Plötz and Stefan Roth. 2018. Neural Nearest Neighbors Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. 1095–1106.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

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

[23] Xiang Zhou, Yuan Zeng, and Yi Gong. 2023. Learning to Scale Temperature in Masked Self-Attention for Image Inpainting. *ArXiv* abs/2302.06130 (2023). https://api.semanticscholar.org/CorpusID:256827540



Cakes Stacked On Each Other

A Shiny White Car

Orange Curtains

Figure 7: Shadows: The images illustrate that the shadows are consistent with the original image before object editing.

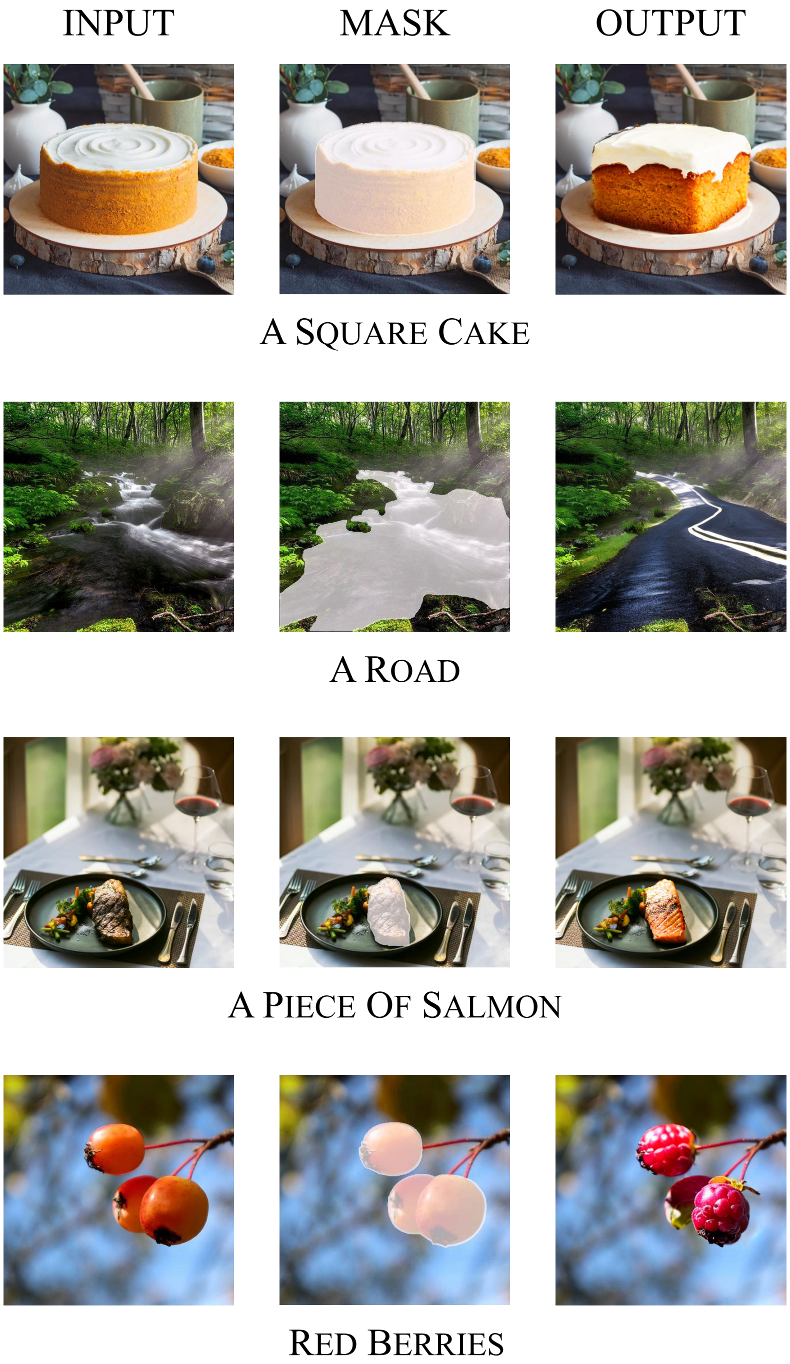


Figure 8: Consistency in illumination of object edit with background: Note the initially illuminated portion of the cake situated on the right, with the relatively dimmer region on the left. Output depiction of the square cake also adheres to the luminosity characteristics inherent in the initial input, thereby ensuring coherence in visual representation. Observe a similar coherence in salmon, road, and red berries.

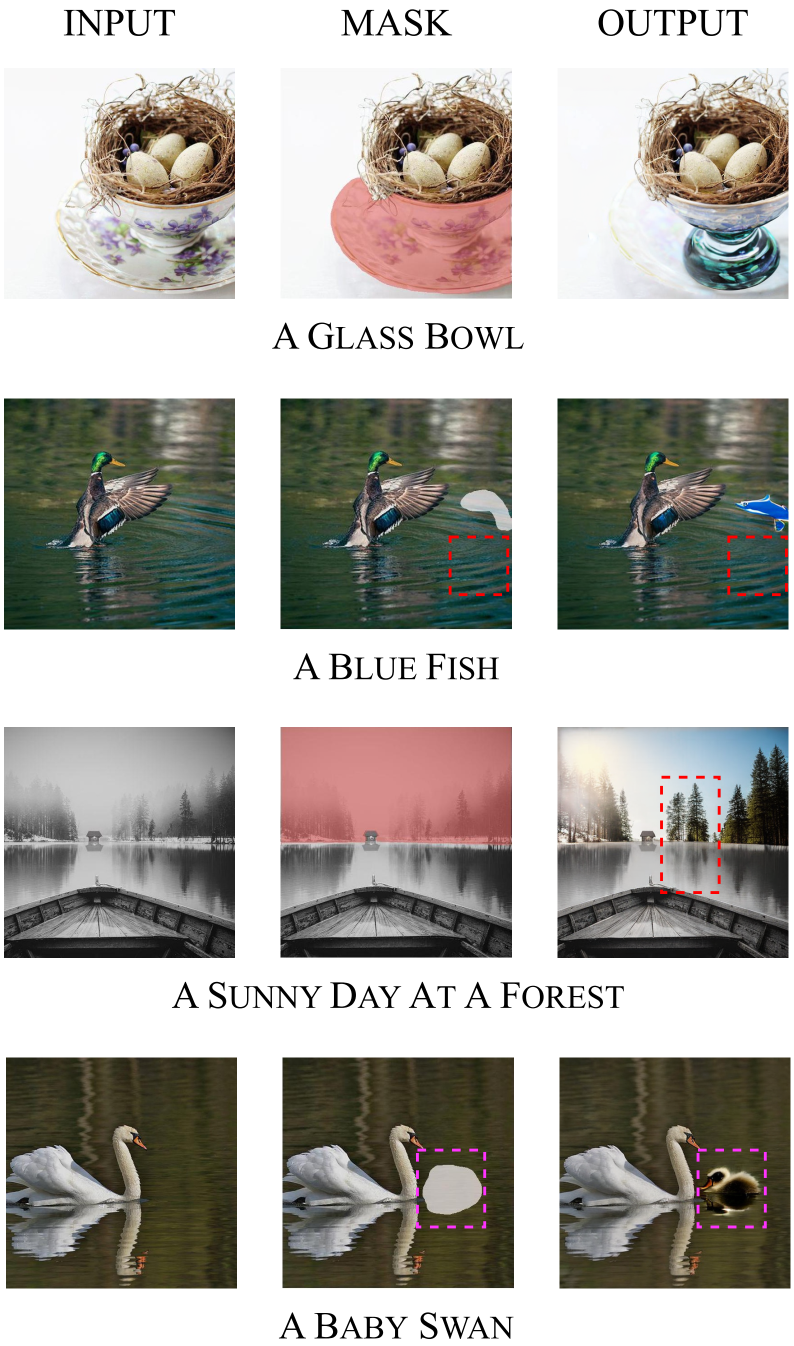


Figure 9: Observe the consistency in reflections of the edited object with respect to the background as highlighted by a dashed rectangle for emphasis.

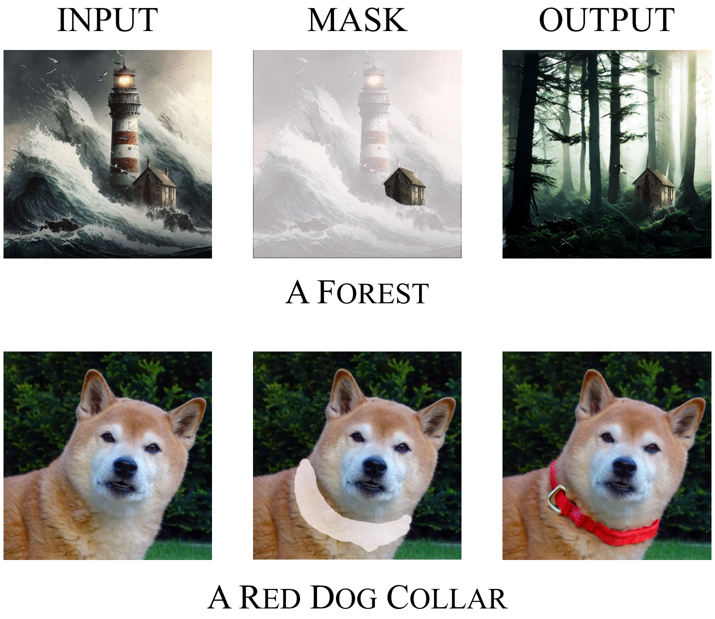


Figure 10: Realism: Note the meticulous attention to detail in the editing of the forest scene, which imbues it with a semblance of reality. Likewise, the dog collar in the output exhibits a naturalistic deformation along the neck region, thereby enhancing realism.

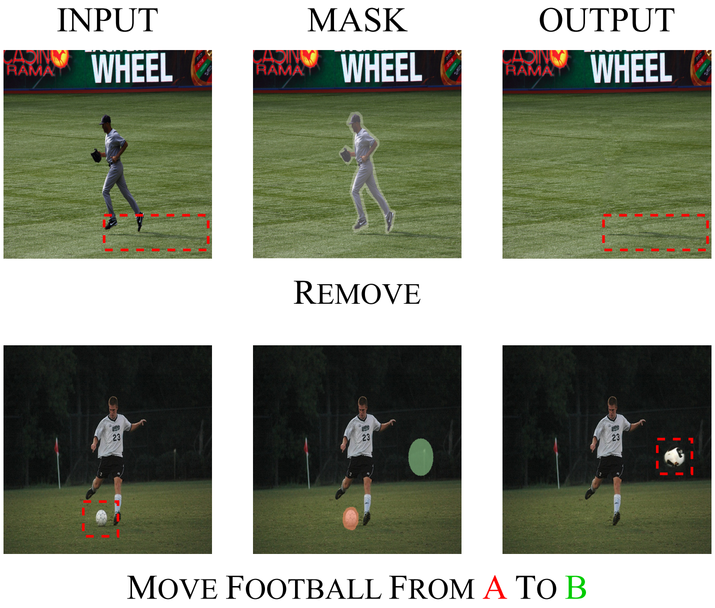


Figure 11: Deletion: LoMOE's limitations about shadow handling. Specifically, upon the removal of an individual, their associated shadows persist post-editing due to our approach's failure to address elements outside the designated mask area. Furthermore, it is noteworthy that despite the presence of motion, the edited football image retains clarity without exhibiting blurriness.

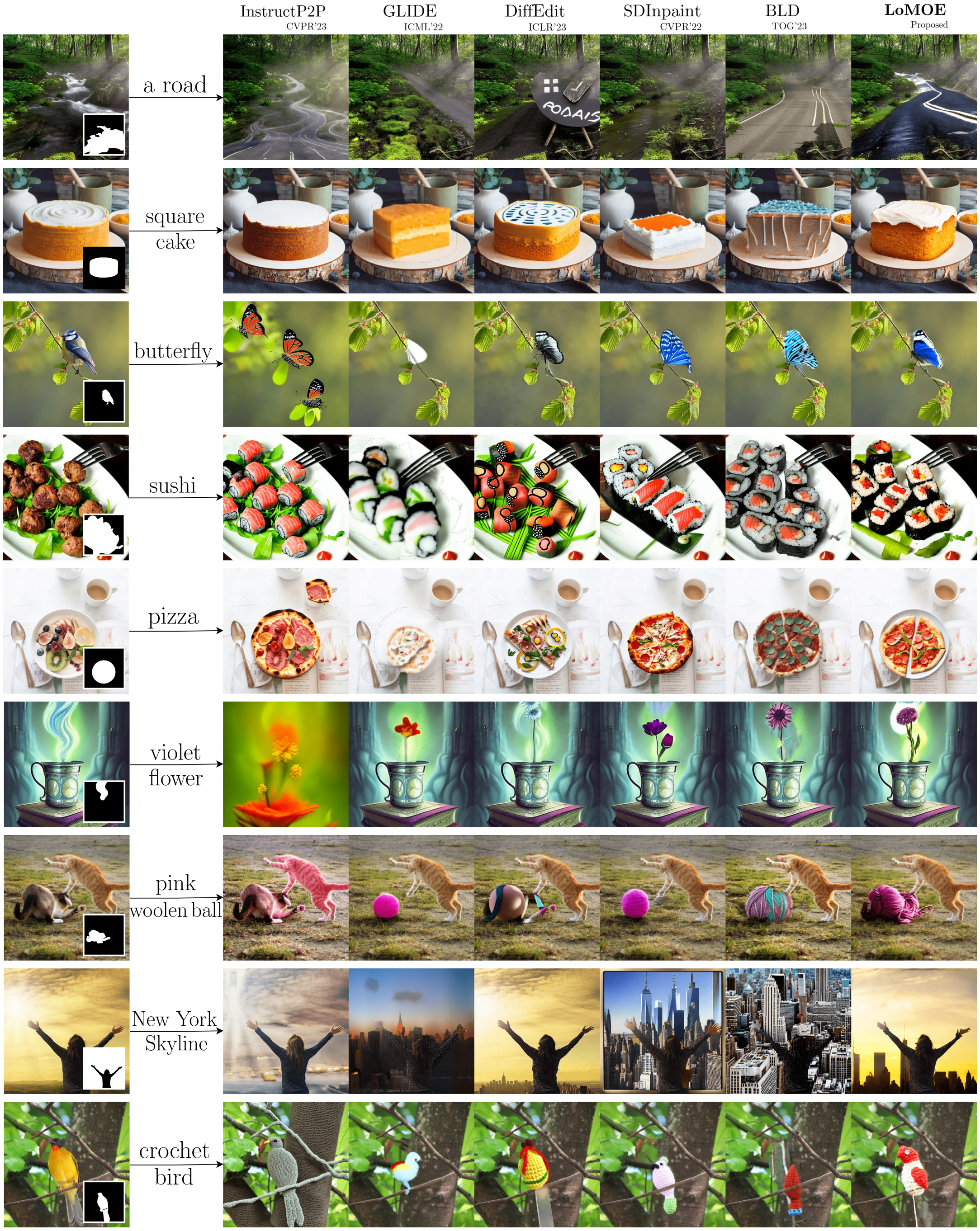


Figure 12: Additional Comparison among Contemporary Methods for Single Object Edits: We present a qualitative comparison of LoMOE against other baseline methods on additional single-object edits. The observations stand similar to Fig. 3 in the main paper, where our proposed method LoMOE makes the intented edit, preserves the unmasked region and avoids unintended attribute edits.

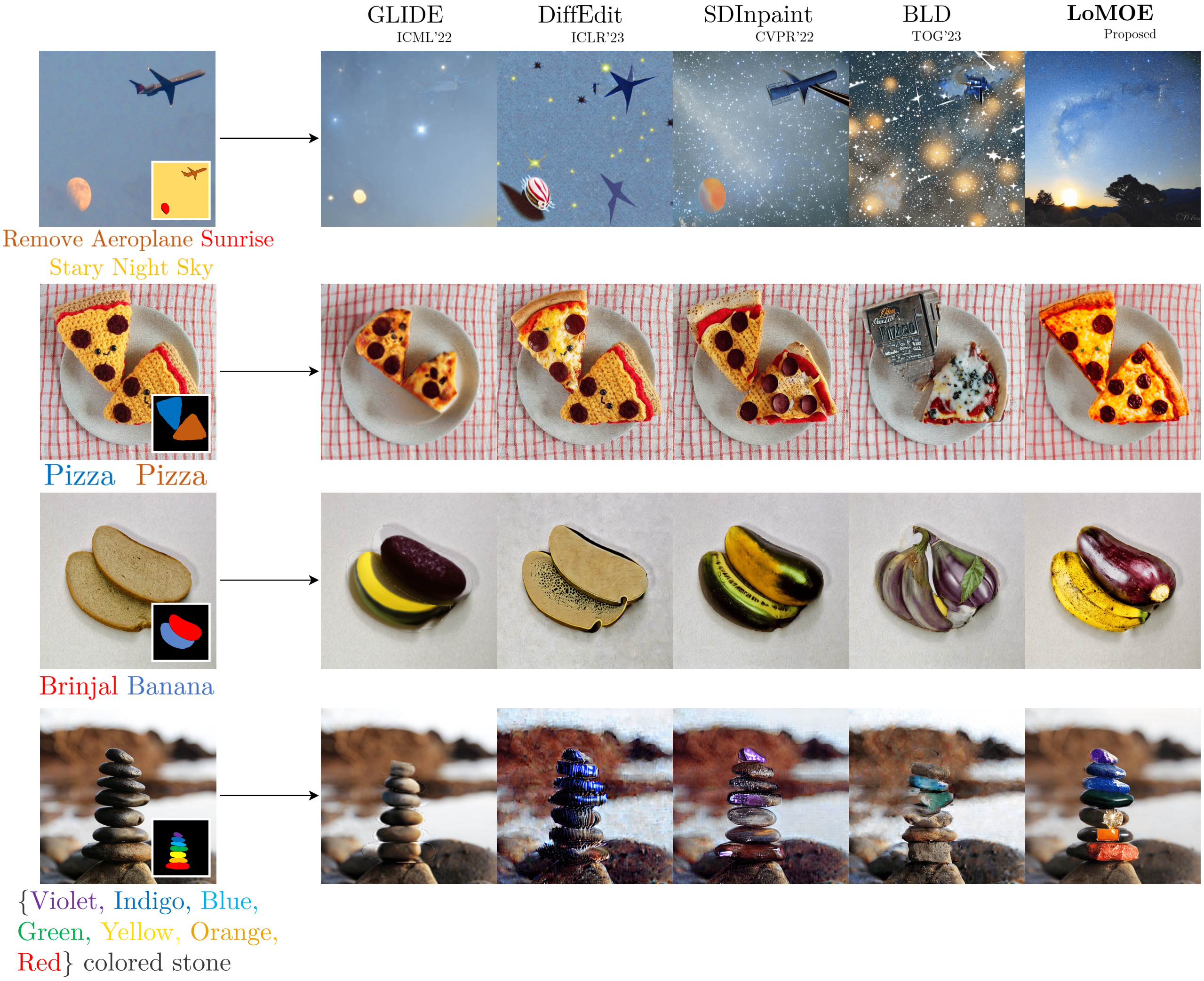
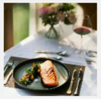
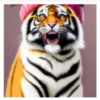
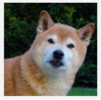
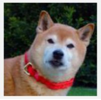


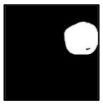
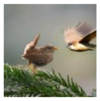
Figure 13: Additional Comparison among Contemporary Methods for Multi Object Edits: We present a qualitative comparison of LoMOE against other baseline methods on additional multi-object edits. The observations stand similar to Fig. 4 in the main paper, where our proposed method LoMOE makes the intended edit, preserves the unmasked region, and avoids unintended attribute edits.

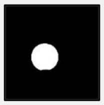
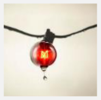
Edit Type	Input Image	Mask	Annotation	Edited image

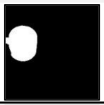
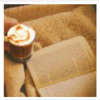
Change Object			{
			TIP: a plate with roasted salmon on it
			SMP: a steak
			TMP: a roasted salmon
			EIn: change the steak to roasted salmon
			}
				

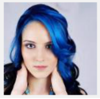
			{
			TIP: a roaring tiger wearing a pink hat
			SMP: a cat
			TMP: a roaring tiger
			EIn: Change the animal from a cat to a roaring tiger
			}
				

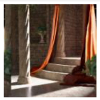
Adding Object			{
			TIP: a dog with a red dog collar looking at the camera
			SMP: no collar
			TMP: red dog collar
			EIn: put a red dog collar on the dogs neck
			}
				

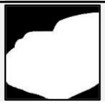
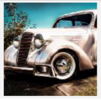
			{
			TIP: a small brown bird sitting on top of a pine tree and a bird flying
			SMP: background
			TMP: a bird flying
			EIn: Add a small bird flying
			}
				

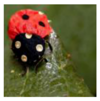
Change Content			{
			TIP: a bulb with red glowing filament hangs from a string
			SMP: filament
			TMP: red glowing filament
			EIn: Add red color to the lights
			}
				

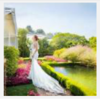
			{
			TIP: a glass of hot chocolate
			SMP: glass of cold coffee
			TMP: glass of hot chocolate
			EIn: change cold coffee to hot chocolate
			}
				

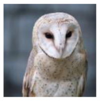
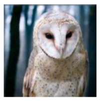
Change Color			{
			TIP: a woman with blue hair
			SMP: brown hair
			TMP: blue hair
			EIn: change brown to blue hair
			}
				

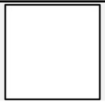
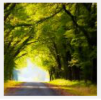
			{
			TIP: a red curtain is hanging over a stone wall
			SMP: red curtain
			TMP: orange curtain
			EIn: change the color of the curtain from red to orange
			}
				

Change Material			{
			TIP: a shiny metal car
			SMP: rusted metal car
			TMP: shiny metal car
			EIn: make the car shiny metal
			}
				

			{
			TIP: a fabric ladybug with black spots on its back is sitting on a leaf
			SMP: a ladybug
			TMP: a fabric ladybug
			EIn: make the ladybug fabric
			}
				

Change Background			{
			TIP: "a bride standing in a garden"
			SMP: on a mountain
			TMP: a garden
			EIn: substitute the mountain with a garden
			}
				

			{
			TIP: a barn owl with a black eye and spots in a blurry forest background
			SMP: a grey background
			TMP: a blurry forest background
			EIn: add a blurry forest background
			}
				

Change Style			{
			TIP: a spring road lined with trees and leaves
			SMP: an autumn road lined with trees and leaves
			TMP: a spring road lined with trees and leaves
			EIn: change the season from autumn to spring
			}
				

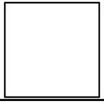
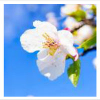
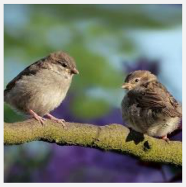
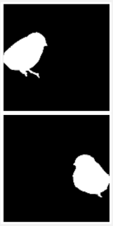
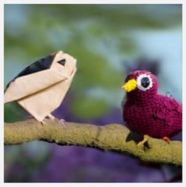
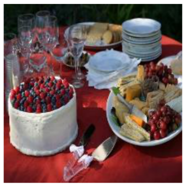
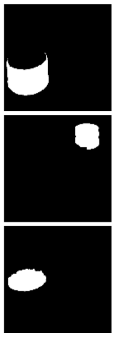
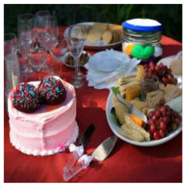
			{
			TIP: an oil painting of a cherry blossom with blue sky
			SMP: cherry blossom with blue sky
			TMP: an oil painting of a cherry blossom with blue sky
			EIn: add an oil painting effect to the cherry blossom
			}
				

Figure 14: Single Object Benchmark: Examples from Single-Object dataset. The columns are (1) Edit type (2) The input image on which the editing is done, (3) The mask used for localizing the edit, (4) JSON annotation containing the Target Image Prompt (TIP), Source Mask Prompt (SMP), Target Mask Prompt (TMP), and the Edit Instruction (EIn), and (5) The edited images produced by LoMOE.

Input Image	Mask	Annotation	Edited image

		{
		SMP: "a brown bird",
		"a brown bird"
		TMP: "a crochet bird",
		"a origami bird"
		}
			

		{
		SMP: "pink cake frosting",
		"a real jar of candy",
		"chocolate cake truffles"
		TMP: "pink cake frosting",
		"a real jar of candy",
		"chocolate cake truffles"
		}
			

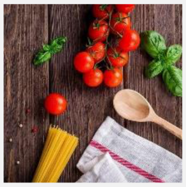
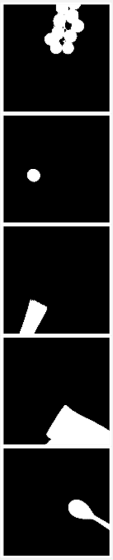
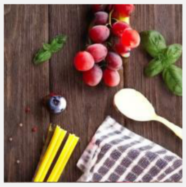
		{
		SMP: "small tomatoes",
		"a small tomato",
		"uncooked spaghetti",
		"a table cloth with red margin",
		"a wooden spoon"
		TMP: "grapes",
		"a blueberry",
		"plastic straws",
		"a checkered table cloth",
		"a steel spoon"
		}
			

Figure 15: LoMOE-Bench: Examples from Multi-Object Dataset. The columns are (1) The input image on which the editing is done, (2) The masks used for localizing the edit, (3) JSON annotation containing the Source Mask Prompts (SMP) and Target Mask Prompts (TMP), and (4) The edited images produced by LoMOE.

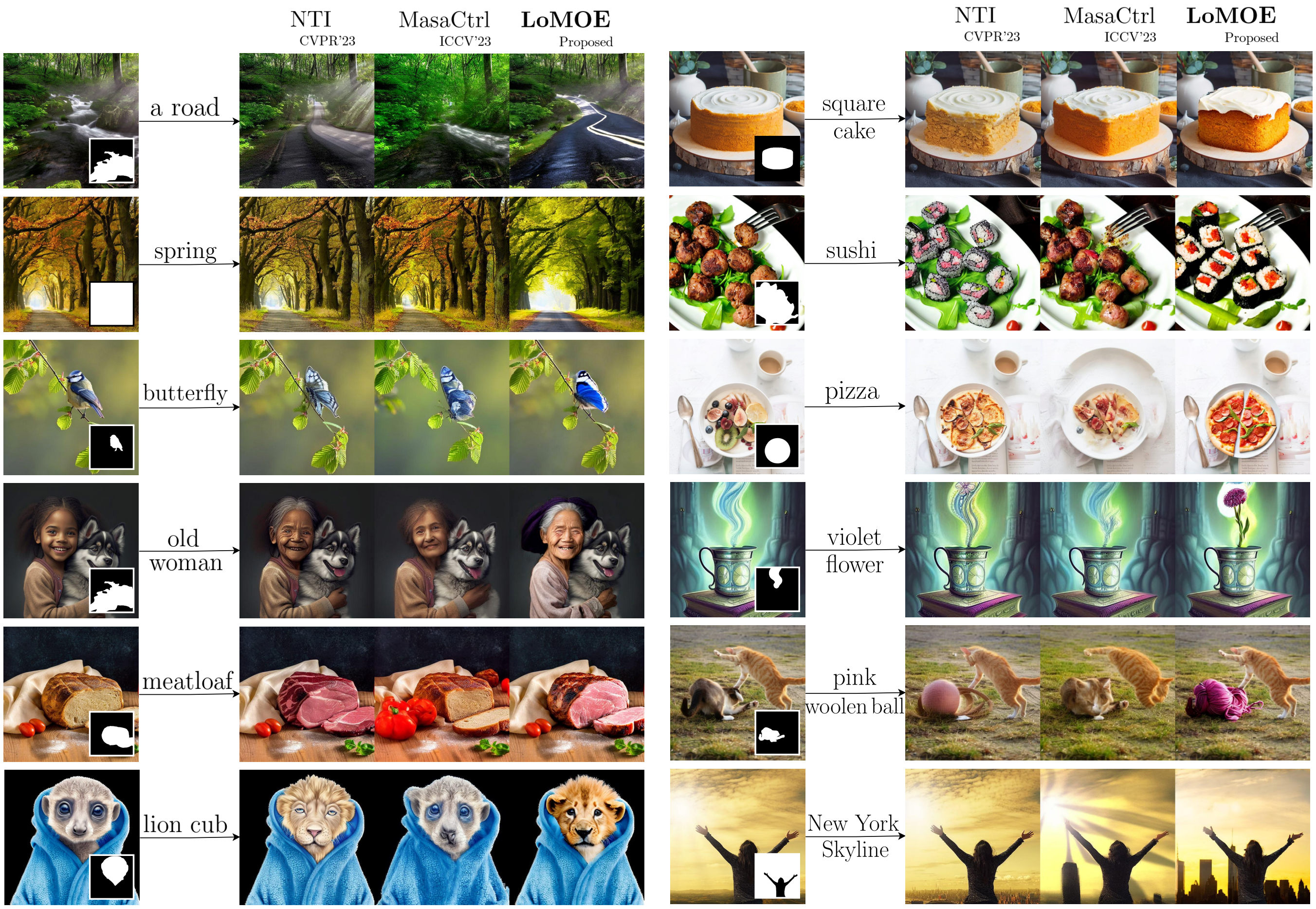


Figure 16: Additional Comparison among Contemporary Methods for Single Object Edits:

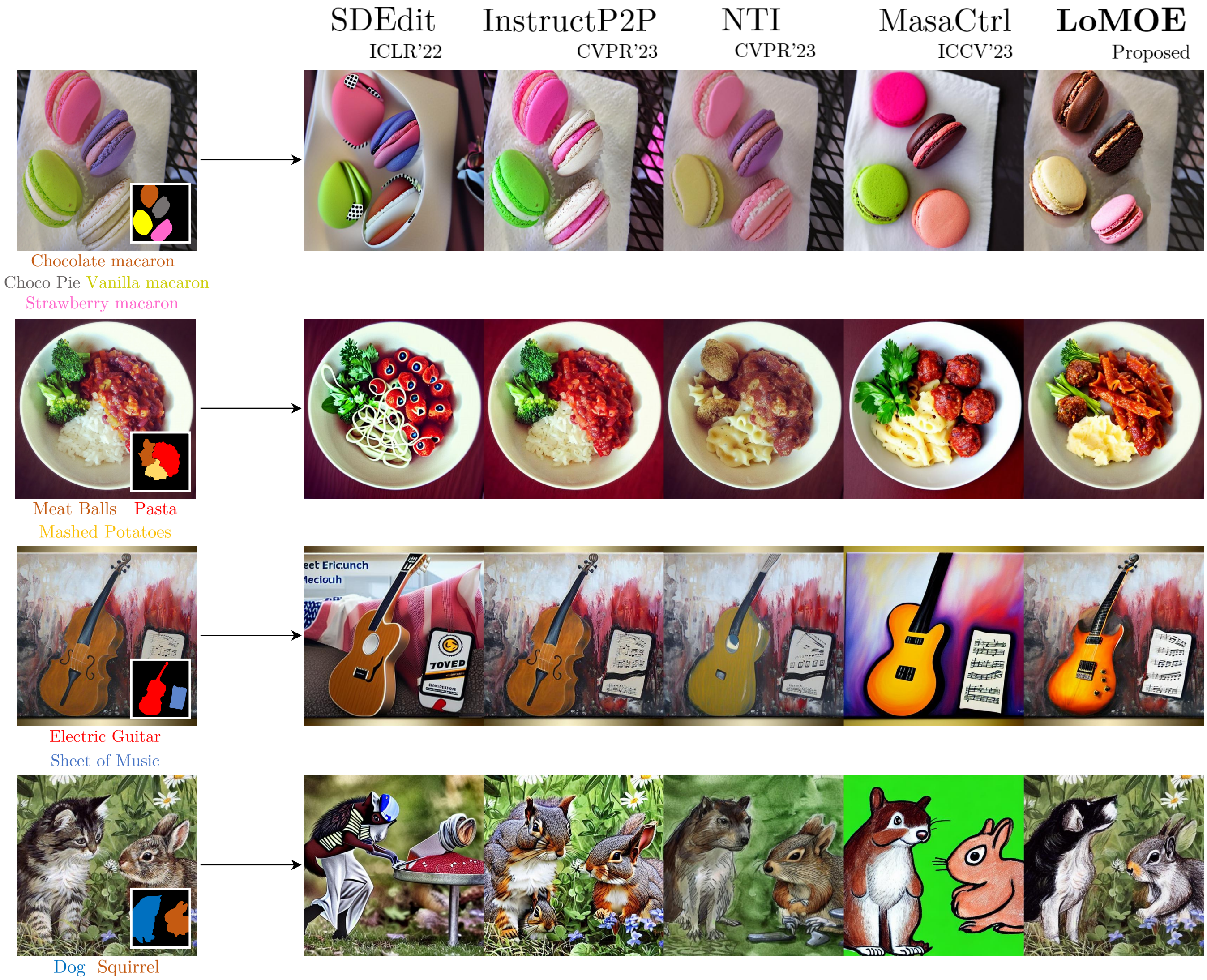


Figure 17: Additional Comparison among Contemporary Methods for Multi Object Edits:

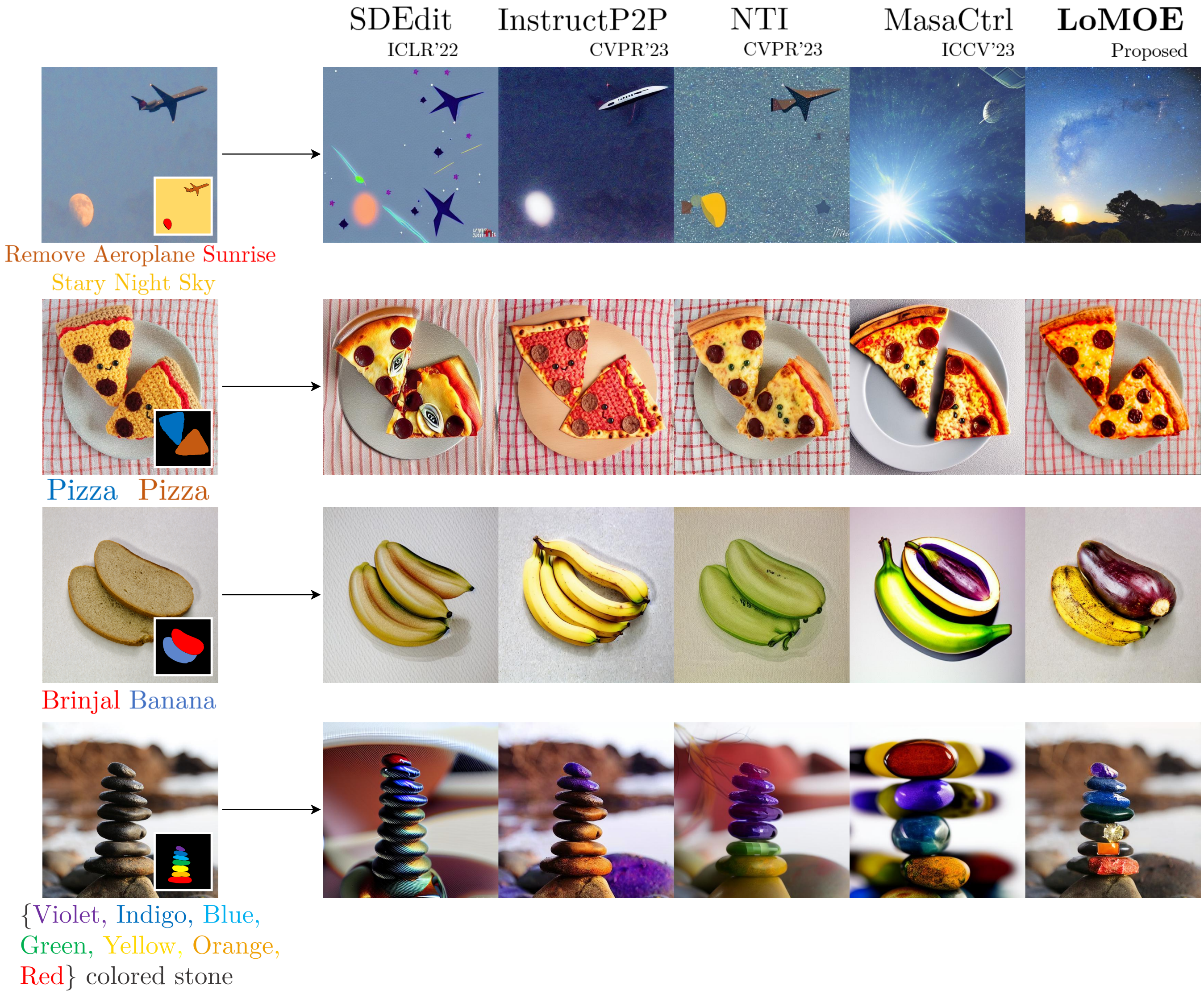


Figure 18: Additional Comparison among Contemporary Methods for Multi Object Edits: