2D Instance Editing in 3D Space

Yuhuan Xie[1] **Aoxuan Pan**[2] **Ming-Xian Lin**[1] **Wei Huang**[1] **Yi-Hua Huang**[1†] **Xiaojuan Qi**[1†]
[1] The University of Hong Kong [2] Independent

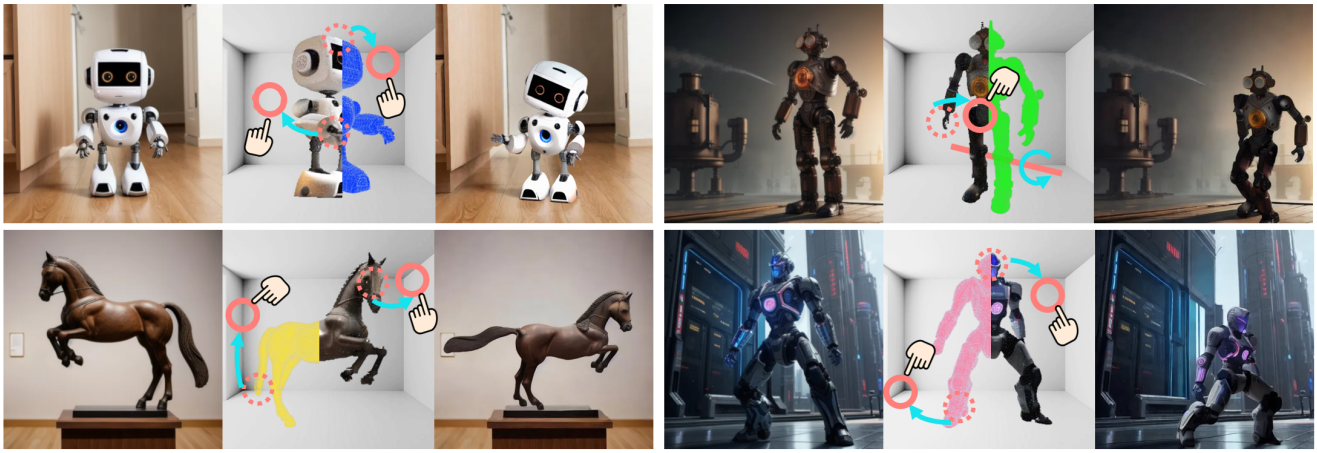

Figure 1. Given 2D instances in images, we propose a method to lift them into 3D, enabling edits under the local rigidity assumption. This approach facilitates large view changes, non-rigid deformations, and interactive, real-time operations for more efficient user control.

Abstract

Generative models have achieved significant progress in advancing 2D image editing, demonstrating exceptional precision and realism. However, they often struggle with consistency and object identity preservation due to their inherent pixel-manipulation nature. To address this limitation, we introduce a novel "2D-3D-2D" framework. Our approach begins by lifting 2D objects into 3D representation, enabling edits within a physically plausible, rigidity-constrained 3D environment. The edited 3D objects are then reprojected and seamlessly inpainted back into the original 2D image. In contrast to existing 2D editing methods, such as DragGAN and DragDiffusion, our method directly manipulates objects in a 3D environment. Extensive experiments highlight that our framework surpasses previous methods in general performance, delivering highly consistent edits while robustly preserving object identity.

1. Introduction

The rapid advancements in deep generative models have fundamentally transformed digital image editing, moving beyond traditional techniques such as filters and color adjustments. These models now provide users with sophisticated tools to manipulate image content with unprecedented precision and realism, enabling direct edits through text-based prompts [4, 32] or reference-based guidance [15, 31]. This evolution has driven significant progress in the field of image editing, propelled by the growing capabilities of generative models. Despite their advancements, most existing approaches rely on high-level semantics derived from language or images as control signals, which often lack the granularity required for precise, fine-grained control over image details. Incorporating interactive controls with intermediate editing results offers a promising solution, enabling real-time feedback that facilitates more accurate adjustments and flexible refinements.

While some recent works, such as DragGAN [22] and DragDiffusion [25], enable interactive image edits by allowing users to pick and drag control points, these methods are constrained to editing a limited range of categories, largely due to the restricted capabilities of their data-driven generative models. Moreover, both DragGAN and DragDif-

†Corresponding Author
Project page:https://panaoxuan.github.io/2D-SpaceEdit-web/.

fusion operate as 2D image models, "manipulating" pixel data rather than addressing the underlying 3D structure of objects. This limitation affects consistency and the preservation of object identity in the edits. We hold that a more intuitive approach to image editing should involve directly manipulating objects in 3D space like a sculpture. This method aligns more closely with human understanding of the 3D physical world, enabling more efficient and effective editing to achieve desired results with greater control and accuracy.

To achieve this goal, we propose a pipeline consisting of several steps: lifting 2D objects to 3D, editing 3D objects under a physically plausible rigidity assumption, positioning the 3D objects, and inpainting them back into the original image. First, we employ SAM [16] to segment the target object in the image. Next, we leverage the state-of-the-art 3D generation method, TRELLIS [29], to produce a 3D Gaussian Splatting (3DGS) [14] representation of the object. This approach is chosen for its real-time rendering capabilities and high visual quality. Using the generated 3DGS, we develop a real-time interactive editing algorithm that deforms the 3DGS under a local rigidity assumption, ensuring structural preservation and physical fidelity. Sparse control points are sampled from the 3DGS point cloud, and neighboring points are connected to construct a topology-aware graph. We then optimize the local rigid energy with editing constraints, which drives the deformation of the dense 3DGS. Once the 3D object has been edited, it is placed back into the image at the specified 6DoF pose (translation + rotation). Finally, the surrounding region is inpainted to ensure the results are visually harmonious.

Our main contributions can be summarized as:

- We propose a novel and comprehensive framework for 3D-aware object editing from a single 2D image. This framework enables large-scale 3D deformation and placement while preserving object identity far better than 2D-based methods.
- We develop a real-time interactive 3DGS algorithm to achieve this goal. By leveraging a rigidity assumption and sparse graph modeling, our method enables real-time editing of 3DGS with physically plausible effects.
- Through extensive experiments, we demonstrate that our approach produces high-fidelity, 3D-consistent results across a wide variety of objects, significantly outperforming prior methods in both realism and user control.

2. Related Work

Our work rethinks 2D image editing by lifting editable object instances into 3D space, enabling geometry-aware manipulation and coherent reintegration within the original scene. We review related works in three interconnected domains: 2D image editing with generative models, lifting 3D geometry from single images, and 3D-aware object editing.

2.1. 2D Image Editing with Generative Models

Classical methods. Traditional image editing methods provided algorithmic tools for tasks such as inpainting, resizing, and compositing. Notable examples include Navier-Stokes inpainting [2], seam carving [3], and Poisson blending [1]. While effective in controlled cases, these methods often fail to capture the semantic or geometric structure of complex scenes.

Learning-based image editing. The introduction of deep generative models brought greater flexibility and realism to image editing. GAN-based frameworks like Pix2Pix [9], CycleGAN [7], and StyleGAN [12] enabled tasks such as image-to-image translation and latent space manipulation. These methods support applications like facial control or style transfer, though they may struggle with fine-grained control and even out-of-distribution generalization. Diffusion models have further advanced editing capabilities, offering improved stability and fidelity. Stable Diffusion [24] and DALL-E 2 allow high-quality synthesis from text prompts, while Prompt-to-Prompt [10] and InstructPix2Pix [4] introduce more localized, controllable edits. ControlNet [33] enhances diffusion-based editing by incorporating spatial conditions such as edges, poses, or depth. Despite their strengths, these models remain fundamentally 2D and lack mechanisms to enforce 3D consistency.

Interactive point-based editing. Recent systems such as DragGAN [23] and its diffusion-based extensions [21, 25] offer interactive control over image content through point pairs. These tools enable users to specify the movement or deformation of an object, yielding visually striking outcomes. However, the manipulation occurs entirely in the image plane, without explicit modeling of depth. Such limitations make it difficult to maintain physical plausibility, particularly for edits involving large viewpoint changes.

Our perspective. Unlike the methods above, our approach lifts object edits into a 3D-aware space. By reconstructing a geometry-consistent representation of the object, we enable edits that align with the underlying structure of the object and preserve coherence during reintegration.

2.2. Single-View Lifting from 2D to 3D Space

Estimating 3D structure from a single image has long been a fundamental problem in computer vision. Early deep learning approaches aimed to recover explicit shapes, such as voxel grids or surface meshes [6]. More recent work has shifted toward implicit neural representations, which tend to generalize better and produce smoother results. The introduction of Neural Radiance Fields (NeRF) [19] marked a turning point for view synthesis, although it originally required multiple views of a scene. Follow-up work gradually adapted NeRF to settings with fewer or even single images. For instance, Zero-1-to-3 [17] uses large-scale diffusion models to generate novel views from just one input image,

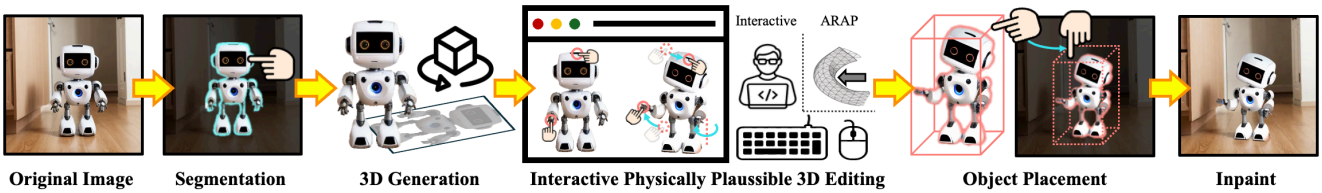

Figure 2. Our method involves lifting 2D instances into 3D space, editing them under the assumption of local rigidity, and then repositioning them back into the desired locations before inpainting them into the images.

capturing a strong sense of object geometry. Meanwhile, 3D Gaussian Splatting [13] provides an efficient and high-fidelity way to represent and render scenes using explicit point-based primitives. These developments have made it feasible to lift a single image into a 3D-aware representation. In our case, this capability is not used for rendering new views, but instead forms the basis for editable object manipulation and seamless re-integration within the original scene.

2.3. 3D-Aware Object Editing

3D-aware editing aims to modify image content in ways that are consistent with the underlying geometry of the scene. One line of research focuses on 3D-aware generative models, such as EG3D [5], which offer explicit control over object pose and appearance. These methods are typically trained on narrow object categories like faces or cars, and their ability to generalize to arbitrary scenes is limited. Another direction explores scene-level editing using volumetric representations. Instruct-NeRF2NeRF [8] allows users to edit a NeRF scene through text prompts, but such edits affect the entire scene rather than individual objects. Our work addresses a different setting. Given an image, we isolate an arbitrary object, lift it into a 3D representation that supports editing, and then reinsert the modified object into the original scene. To restore occluded regions, we leverage powerful modern inpainting models [28], allowing the final composite to remain coherent and visually plausible. This object-centric, compositional approach enables geometry-aware editing of in-the-wild images without requiring category-specific priors or multi-view input.

3. Method

The pipeline of our method, as illustrated in Fig. 2, comprises the following steps: instance segmentation, 2D-to-3D generation, 3D editing, and inpainting the rendered edits of 3D shapes back into the image.

3.1. Lifting 2D Instances to 3D Space

Unlike existing image-based editing techniques that manipulate pixels in 2D space using generative models such as GANs or Diffusion, our approach focuses on editing 2D instances by crafting them as 3D objects. This method provides a more intuitive and human-aligned understanding of the physical world. The process begins by identifying the instances to be edited and lifting them into 3D space as a 3D representation.

Aiming at this goal, given an image containing instances to edit, we first employ SAM [16] to isolate each individual. The resulting segments are then cropped to center these instances. These cropped images are subsequently processed using the off-the-shelf 3D generation model TRELLIS [29], which produces the 3D Gaussian Splatting (3DGS) [14] representation, chosen for its fast rendering speed and high visual quality.

The obtained 3DGS is represented as a set of Gaussian kernels $\mathcal{G}_i : (\mu_i, o_i, s_i, q_i, c_i)$, where $\mu_i \in \mathbb{R}^3$ denotes the center of the Gaussian, $o_i \in \mathbb{R}^+$ represents the opacity, $s_i \in \mathbb{R}^{+3}$ corresponds to the scaling factors along the 3D axes, and $q_i \in \mathbb{R}^4$ is the quaternion representing the $SO(3)$ rotation of the Gaussian.

3.2. Physics-Sensible Interactive Editing

With the 3DGS representation $\mathcal{G}_i : (\mu_i, o_i, s_i, q_i, c_i)$ in hand, the next step is to efficiently edit it while ensuring adherence to physically plausible constraints. To achieve this, we adopt the concept of ARAP [11, 26], enabling interactive editing of 3DGS while maintaining the physical prior that each local part of the 3D shape preserves as much rigidity as possible. This approach applied to 3DGS ensures that the edited results undergo plausible deformations while strictly adhering to user-defined constraints.

Since 3DGS contains millions of Gaussian primitives to accurately approximate the target object, performing deformation based on per-Gaussian analysis is computationally prohibitive for real-time interaction and memory efficiency. To address this, we derive a set of coarse control points evenly distributed over the 3D shape using farthest-point sampling on the Gaussians. During our experiments, the number of control points is set to 512. Denoting control point positions before deformation as p_i and their deformed positions as p'_i, we define the local rigidity energy as:

$$E(p') = \sum_i \sum_{j \in \mathcal{N}i} w_{ij} |(p'_i - p'_j) - R_i(p_i - p_j)|^2, \quad (1)$$

where R_i is the estimated rotation of the local cell centered at p_i, $\mathcal{N}i$ is the neighborhood of p_i consisting of

$K = 8$ nearest neighbors, and wij represents the importance weight of p_j relative to p_i. This energy term is both translation- and rotation-invariant, penalizing any local non-rigid deformation. Rather than directly applying K-nearest neighbors (K-NN) in Euclidean space, we first use $\frac{K}{2}$-nearest neighbors to connect control nodes and construct an initial graph. Then, we compute the K-NN based on the shortest path distances within this graph. As illustrated in Fig. 3, this approach preserves the topology of the final connected graph by preventing control nodes from linking to overly distant points from separate regions.

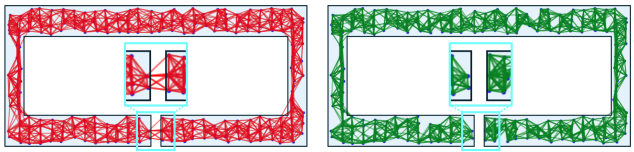

Figure 3. Graph better preserves topology than the Euclidean.

When the user interacts with the system by clicking on the image and dragging H selected control points to reposition them in 3D space, we impose strict constraints on these handle points:

$$p'_{h_i} = \tilde{p}_{h_i}, \quad i \in [H], \tag{2}$$

where h_i denotes the indices of the selected control points.

The term in Eq. 1 can be interpreted as penalizing deviations in relative positions, centralized by the Laplacian matrix and rotated by the estimated local rotation. This can be expressed as:

$$Lp' = b, \tag{3}$$

where $b = [\sum_{j \in \mathcal{N}i} \frac{w_{ij}}{2}(R_i + R_j)(p_i - p_j)]^T$ is a constant vector, and L is the Laplacian matrix, which centralizes positions relative to their weighted neighboring centers. The minimization of Eq. 3 can be solved linearly using the inverse or pseudo-inverse of L, depending on the constraint setup. The rows and columns corresponding to h_i in L and the rows in b are removed to solve for the unconstrained points, while the constrained points remain fixed at user-defined positions.

The local rotation R_i is estimated using Singular Value Decomposition (SVD) on the matrix [27]:

$$S_i = \sum_{j \in \mathcal{N}i} w_{ij}(p_j - p_i)^T(p'_j - p'_i). \tag{4}$$

By performing SVD on $S_i = U_i \Sigma_i V_i^T$, the local rotation R_i is computed as $R_i = V_i U_i^T$.

The entire solving process is performed iteratively by alternating between optimizing p' and updating R_i, gradually converging toward a global optimum [20]. In practice, three iterations are sufficient to achieve satisfactory results.

Finally, the deformation of Gaussians is driven by their neighboring control points using a Linear Blend Skinning (LBS) [18] approach. We express the deformation of Gaussian parameters as:

$$\begin{aligned} \mu'_i &= \sum_{j \in \tilde{\mathcal{N}}_i} \tilde{w}_{i,j} \left(R_j \left(\mu_i - p_j \right) + p'_j \right), \\ q'_i &= \sum_{j \in \tilde{\mathcal{N}}_i} \tilde{w}_{i,j} \text{Quat} \left(R_j \cdot \text{Rot}(q_i) \right). \end{aligned} \tag{5}$$

Here, $\tilde{\mathcal{N}}_i$ represents the $\tilde{K} = 3$ nearest control points to the i-th Gaussian, and $\tilde{w}_{ij}$ are weights that decay with distance. This method ensures smooth, physically plausible deformations of the Gaussian representation while respecting user-defined constraints.

3.3. 3D Object Placing and Inpainting

The final crucial step in our pipeline is to seamlessly reintegrate the edited 3D object back into the 2D image. The drop stage involves two concurrent processes: restoring the background occluded by the original object and rendering the edited object for final composition. The goal is to produce a coherent and photorealistic image of the manipulated instance placed in the original image.

Lifting the target object from the image inevitably leaves a "hole" in the background where the object was previously located. To create a plausible scene for re-integration, this occluded area must be semantically restored. We accomplish this using the initial segmentation mask generated by SAM. This mask precisely defines the region requiring inpainting. We then employ a state-of-the-art resolution-robust inpainting network to fill the designated area. The network takes the masked image as input and synthesizes the missing background content, ensuring contextual and structural consistency with the surrounding pixels. This process results in a complete and clean background plate, ready to receive the edited object.

Concurrently, the manipulated 3D Gaussian Splatting(3DGS) model is rendered back into a 2D image from a consistent viewpoint. The final composition begins with a standard alpha blending of the rendered object onto the inpainted background. To eliminate any unnatural seams at the object's boundary, we perform a targeted inpainting-based refinement. A dilated mask is generated around the object's contour to isolate the transition area. This masked region is then inpainted using prompts like "seamless integration" to guide the model. This technique effectively harmonizes the boundary by synthesizing a contextually aware transition, ensuring that the final image is photorealistic and coherent. We utilize PixelHacker [30] for inpainting the composed images, given its robust and exceptional performance.

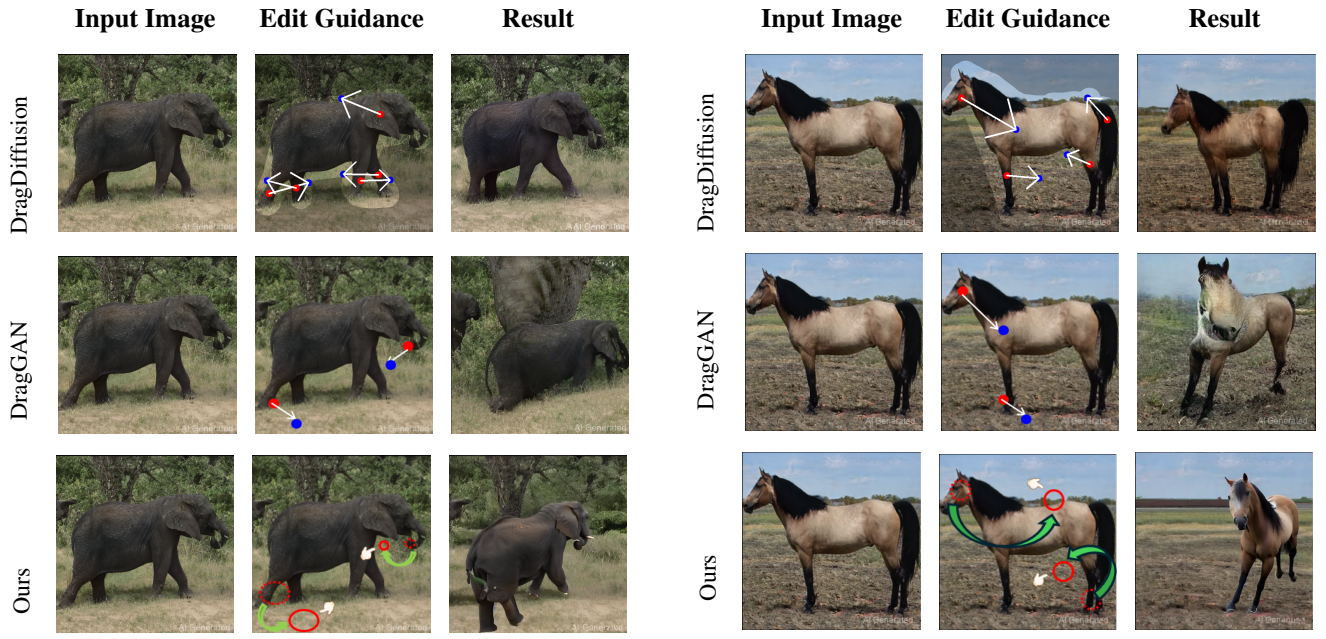

Figure 4. Qualitative comparisons of our method with DragDiffusion [25] and DragGAN [22] on natural images.

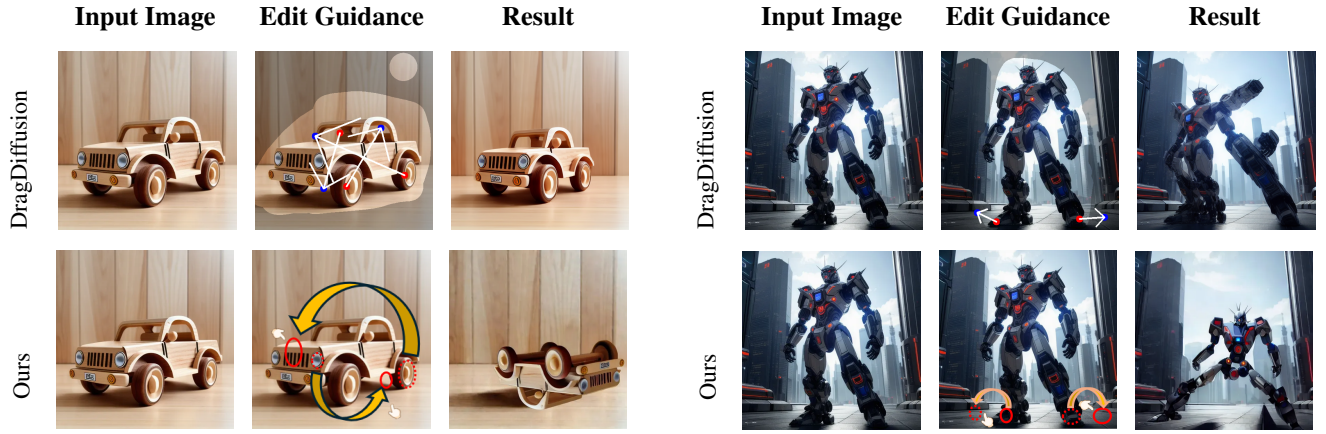

Figure 5. Qualitative comparisons of our method with DragDiffusion [25] on man-made objects. It is noteworthy that DragGAN [22] failed to produce any results for these input images.

4. Result

4.1. Qualitative Comparison

We evaluated our proposed method against two state-of-the-art interactive editing approaches: DragDiffusion and DragGAN. These methods represent the current paradigm of point-based image manipulation, where users specify source and target positions to guide the editing process. Our comparison specifically focuses on editing scenarios that involve significant pose changes or viewpoint modifications. These transformations are particularly challenging for purely 2D-based approaches due to their inherent pixel-level manipulation, but they are naturally handled by our 3D-aware framework.

Our experiments were conducted on both natural images and images of man-made objects. As shown in Fig.4, for natural images of specific targets such as elephants and horses, both DragDiffusion[25] and DragGAN [22] produce edited results, but these are not strictly aligned with the given controls. For instance, the back legs and nose of the elephant cannot be moved forward and backward, respectively. DragDiffusion failed to rotate the horse's face toward the camera due to its in-plane nature. While DragGAN successfully rotated the horse's head, it produced unnatural results on the horse's face.

For images of more general man-made objects (Fig. 5),

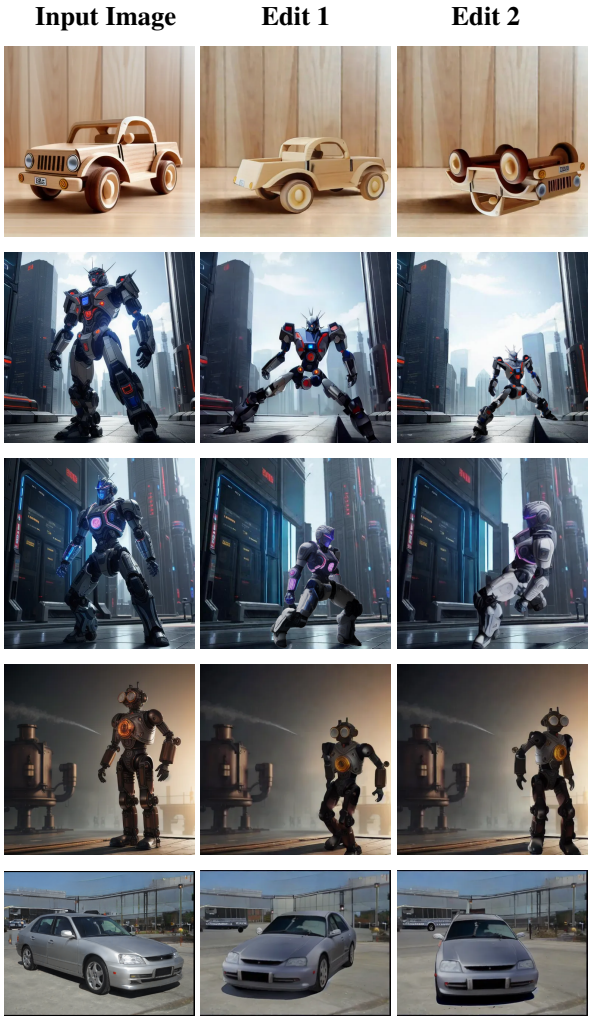

Figure 6. Diverse editing results produced by our method.

we observed that DragGAN failed to produce any results. Meanwhile, DragDiffusion was unable to achieve large rigid rotations of the target objects. Furthermore, DragDiffusion introduced semantic artifacts when handling significant non-rigid deformations, such as on the Gundam legs.

In contrast, our method consistently performs large-scale editing across a wide variety of images, producing high-fidelity outputs that robustly preserve object identity and scene consistency. This comprehensive evaluation demonstrates that our approach significantly outperforms the compared methods in challenging editing scenarios with flexible, non-rigid, accurate, and large-scale 3D control.

4.2. Diverse Editing Results

Our experiments extended to various other images, subjected to diverse editing manipulations. The showcased results in Figure 6 confirm that our method not only supports the large-scale edits users anticipate but also robustly preserves object identity. Qualitative evaluations highlight a significant reduction in visual artifacts and superior structural coherence compared to existing 2D methods, validating our 2D-3D-2D approach's efficacy.

5. Conclusion

Our work presents a real-time interactive 2D image editing method that performs manipulations on 2D instances in 3D space. This framework significantly outperforms prior interactive approaches, e.g, DragGAN and DragDiffusion, enabling large-scale edits across a wide range of images while consistently preserving object identity. However, the fidelity of our edits heavily depends on the quality of 3D reconstruction, which remains challenging for inputs with occlusions, severe lighting conditions, or distant object placement. Moreover, existing off-the-shelf inpainting methods fail to consistently produce satisfactory results. Using the composed images as a coarse draft, training a refinement model to specifically post-process these images is left as future work to achieve higher visual quality and better harmony between the instances and environments.

Acknowledgement

This work has been supported in part by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant No. 17202422, 17212923), Theme-based Research (Grant No. T45-701/22-R) and Shenzhen Science and Technology Innovation Commission (SGDX20220530111405040). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

References

[1] Mahmoud Afifi and Khaled F Hussain. Mpb: A modified poisson blending technique. *Computational Visual Media*, 1:331–341, 2015. 2

[2] Wilson Au and Ryo Takei. Image inpainting with the navier-stokes equations. *Available at Final Report APMA*, 930, 2001. 2

[3] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 609–617. 2023. 2

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 1, 2

[5] Eric R Chan, Connor Z Lin, Matthew A Chan, Matthew Tancik, Eric Ng, Tero Aumentado-

Armstrong, and Avinash Kulkarni. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 3

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part VIII 14*, pages 628–644. Springer, 2016. 2

[7] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 2

[8] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 3

[9] Joyce Henry, Terry Natalie, and Den Madsen. Pix2pix gan for image-to-image translation. *Research Gate Publication*, pages 1–5, 2021. 2

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[11] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 3

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

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 2, 3

[15] Kangyeol Kim, Sunghyun Park, Junsoo Lee, and Jaegul Choo. Reference-based image composition with sketch via structure-aware diffusion model. *arXiv preprint arXiv:2304.09748*, 2023. 1

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2, 3

[17] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023. 2

[18] Nadia Magnenat-Thalmann, Richard Laperrière, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface'88*, pages 26–33, 1989. 4

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021. 2

[20] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996. 4

[21] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 2

[22] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Kripasindhu Aberman, Christian Theobalt, and Jaakko Lehtinen. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 1, 5

[23] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 2

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[25] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024. 1, 2, 5

[26] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116. Citeseer, 2007. 3

[27] Olga Sorkine-Hornung and Michael Rabinovich.

Least-squares rigid motion using svd. *Computing*, 1 (1):1–5, 2017. 4

[28] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 3

[29] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 3

[30] Ziyang Xu, Kangsheng Duan, Xiaolei Shen, Zhifeng Ding, Wenyu Liu, Xiaohu Ruan, Xiaoxin Chen, and Xinggang Wang. Pixelhacker: Image inpainting with structural and semantic consistency. *arXiv preprint arXiv:2504.20438*, 2025. 4

[31] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023. 1

[32] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 1

[33] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2