3DEgo: 3D Editing on the Go! (Supplementary Material)

Umar Khalid[1,*], Hasan Iqbal[2,*], Azib Farooq[3], Jing Hua[2], and Chen Chen[1]

[1] University of Central Florida, Orlando, FL, USA
[2] Department of Computer Science, Wayne State University, Detroit, MI, USA
[3] Department of Computer Science and Software Engineering, Miami University, Oxford, OH, USA

A Overview

The supplementary material has been organized into the following sections:

- Section B: Implementation Details
- Section C: Additional Results
- Section D: User Study
- Section E: Autoregressive Editing with Diffusion Models
- Section F: Broader Impact
- Rendered videos of the edited scenes are provided together with the PDF document. Due to the file size limitation imposed by ECCV, we had to compress some of the video samples extensively, which may have impacted their quality.

B Implementation Details

In the development of our method, we leverage PyTorch [12] with a specific focus on 3D Gaussian splatting techniques. The process of determining the key editing areas (KEAs) utilizes GPT-3.5 Turbo [3] to pinpoint the relevant editing attributes. SAM [10] is deployed for mask generation, aiming at isolating the KEAs. For the task of zero-shot point tracking, our approach incorporates a point-tracker as delineated in [13]. Editing operations are executed with the aid of the Instruct Pix2Pix [2] 2D diffusion model, which applies the generated masks to confine edits within the designated KEAs. For instance, given an editing prompt, "Make bulldozer red and give cones yellow color". GPT [3] should output "bulldozer" and "cones" as the editing attributes and SAM [10] should generate masks for these objects.

For 3D reconstruction, we adhere to the optimization parameters and configurations established in 3DGS [8], except where noted. We synchronize the addition of new frames with point densification intervals to ensure the scene's progressive expansion. Camera poses are refined using quaternion rotation $\mathbf{q} \in \mathfrak{so}(3)$

* Equal Contribution

Table 1: Comparing With View-Consistent NeRF Editing Methods. Quantitative evaluation of 200 edits across GS25, IN2N, Mip-NeRF, NeRFstudio, Tanks & Temples, and CO3D-V2 datasets against the methods that incorporate COLMAP poses. Every baseline approach is trained using its publicly available code following the original configurations and assessed using a uniform evaluation protocol. The top-performing results are emphasized in bold.

Datasets	ViCA-NeRF [5]			CSD [9]			Ours		
	CTIS↑	CDCR↑	E-PSNR↑	CTIS↑	CDCR↑	E-PSNR↑	CTIS↑	CDCR↑	E-PSNR↑
GS25 (Ours)	0.147	0.842	21.513	0.135	0.848	21.874	**0.169**	**0.925**	**23.660**
Mip-NeRF	0.142	0.851	22.727	0.156	0.871	21.081	**0.175**	**0.901**	**24.250**
NeRFstudio	0.148	0.858	22.489	0.163	0.864	23.874	**0.163**	**0.931**	**24.990**
CO3D-V2	0.165	0.870	23.641	0.155	0.879	23.922	**0.179**	**0.936**	**26.020**
IN2N	0.159	0.876	23.542	0.171	0.866	25.189	**0.183**	**0.925**	**26.390**
Tanks & Temples	0.143	0.851	22.823	0.162	0.859	21.974	**0.164**	**0.915**	**24.190**

and translation vector $\mathbf{t} \in \mathbb{R}^3$. The training process begins with an initial learning rate of 50^{-6}, which is gradually reduced to 10^{-6} as the model approaches convergence.

For 2D editing, we use $w = 2$ by default and the scale factors s_f and s_t are adjusted within the intervals $\{1.0, 1.5\}$ and $\{6.5, 12.5\}$, respectively, tailored to the specific requirements of each scene. The parameters γ_f and γ_E are fixed at 0.8 and 0.2, respectively. Throughout both the initialization and the global expansion phases, the weights λ_{BCE} for the Binary Cross-Entropy loss and λ_{JSD} for the Jensen-Shannon Divergence are uniformly maintained at 0.5. The weight λ_{KEA}, relevant to the Key Editing Areas, is assigned values of 0.2 during the initialization phase and adjusted to 0.1 in the expansion phase. The iteration count I varies between 150 and 200, depending on scene characteristics, while the weights λ_{rgb}, λ_{pc}, and λ_{ipc} for RGB, point cloud, and interpolated point cloud losses are set to 1, 0.7, and 0.3, respectively.

C Additional Results

This section delineates the quantitative outcomes derived from extensive evaluations of our method against recent advancements in view-consistent editing and concurrent works published on arXiv. We meticulously assessed the performance of our approach across a spectrum of datasets, including GS25, IN2N, Mip-NeRF, NeRFstudio, Tanks & Temples, and CO3D-V2. These evaluations encompass a total of 200 edits, offering a comprehensive analysis of our method's efficacy relative to state-of-the-art NeRF methods that leverage COLMAP poses and contemporary Gaussian Splatting techniques.

Table 1 presents a comparative analysis against recently accepted papers at NeurIPS 2024, focusing on view-consistent editing. The comparison highlights our method's superior performance across multiple benchmarks. Notably, our approach excels in CTIS, CDCR, and E-PSNR metrics, outperforming notable methods such as ViCA-NeRF [5] and CSD [9]. This excellence underscores our

Table 2: Comparing With Pose-known Gaussian Splatting 3D Editing Methods. Quantitative evaluation of 200 edits across GS25, IN2N, Mip-NeRF, NeRFstudio, Tanks & Temples, and CO3D-V2 datasets against the methods that incorporate COLMAP poses. Every baseline approach is trained using its publicly available code. The top-performing results are emphasized in bold. It must be noted that our method doesn't incorporate any COLMAP poses.

Datasets	GaussianEditor [4]			Gaussian Grouping [15]			Ours		
	CTIS↑	CDCR↑	E-PSNR↑	CTIS↑	CDCR↑	E-PSNR↑	CTIS↑	CDCR↑	E-PSNR↑
GS25 (Ours)	0.162	0.930	**23.888**	0.128	0.803	20.817	**0.169**	**0.925**	23.660
Mip-NeRF	0.156	0.894	25.116	0.148	0.825	19.953	**0.175**	**0.901**	**24.250**
NeRFstudio	**0.164**	**0.947**	24.037	0.154	0.818	22.617	0.163	0.931	**24.990**
CO3D-V2	**0.183**	0.924	26.124	0.147	0.832	22.662	0.179	**0.936**	**26.020**
IN2N	0.175	**0.925**	**27.019**	0.161	0.819	23.859	**0.183**	0.915	26.390
Tanks & Temples	**0.170**	0.896	25.069	0.153	0.811	20.799	0.164	**0.915**	**24.190**

method's robustness and efficiency in generating high-fidelity, view-consistent edits across diverse scenes.

Our evaluation extends to concurrent works on arXiv, where we continue to demonstrate our method's prowess, as summarized in Table 2. Despite the competitive landscape, our method maintains leading performance indicators, particularly against GaussianEditor [4] and Gaussian Grouping [15]. The results underscore our approach's capability to achieve high-quality editing outcomes, reinforced by our advancements in CTIS, CDCR, and E-PSNR metrics.

The quantitative evaluations elucidate our method's leading-edge performance, establishing a new benchmark in the domain of 3D scene editing. Through meticulous refinement and innovative techniques, we push the boundaries of what is achievable, paving the way for future explorations in this rapidly evolving field.

C.1 Additionl Ablations

In an additional ablation study, we explore the impact of varying the number of adjacent views (w) incorporated in our autoregressive editing approach. This parameter w signifies the breadth of context utilized during the editing process, directly influencing the consistency and quality of the edited views. As detailed in the table, our findings indicate a nuanced balance between editing performance and computational cost.

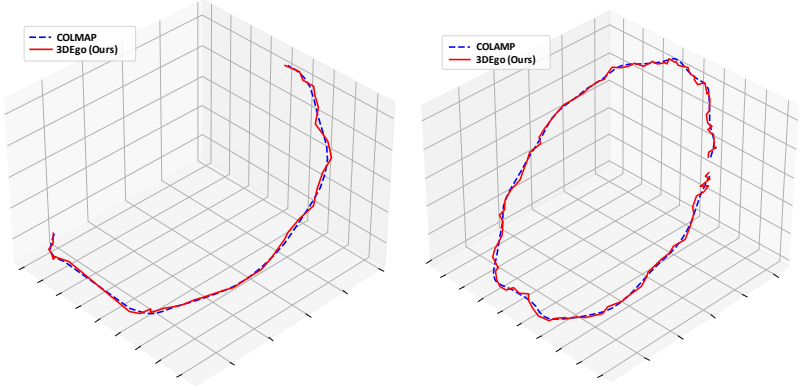
Table 3: An ablation with the number of adjacent views, w. $w > 2$ gives a slightly better performance with an increase in computational cost and editing time

w	CTIS	CDCR	E-PSNR
0	0.134	0.843	21.64
1	0.156	0.865	21.98
2	0.164	0.915	24.190
3	0.166	0.920	24.263
4	0.167	0.923	24.304

With $w = 0$, implying no adjacent views are considered, the model operates with minimal contextual information, leading to baseline performance levels in terms of CTIS, CDCR, and E-PSNR.

As w increases to 1 and 2, we observe substantial improvements in all metrics, underscoring the value of incorporating adjacent views to enhance editing consistency and detail preservation. Specifically, at $w = 2$, we achieve notable enhancements, marking a significant improvement over having no adjacent view context.

Further increments to w (i.e., 3 and 4) continue to marginally elevate performance metrics, indicating diminishing returns on further contextual integration. This slight improvement comes at the cost of increased computational demand and editing time, suggesting an optimal balance at $w = 2$ for efficient and high-quality editing. Based on these observations, we select $w = 2$ as the default setting in our experiments.



(a) Pose estimation for the GS25 Statue **(b)** Pose estimation for the GS25 Van

Fig. 1: Qualitative comparison for camera pose estimation. For visualization, we pick two scenes from GS25 dataset, and compare the poses calculated by our method against the COLMAP poses.

C.2 Pose Estimation

The derived camera poses undergo a Procrustes analysis refinement step, as delineated in [1, 11], prior to their evaluation against the COLMAP poses from the training datasets. We present a qualitative analysis comparing the estimated poses of edited frames to the COLMAP poses in Figure 1. From our GS25 dataset, we selectively examine two distinct scenes: one captured with a 360-degree camera movement (Van scene) and another observed from a 180-degree viewpoint (Statue scene).

In evaluating the camera pose estimation performance of our approach further, we employ standard metrics used in the field, including the Absolute Trajectory Error (ATE), Relative Pose Error (RPE) for translation (RPE_t), and RPE for rotation (RPE_r), following the methodologies proposed in [1, 11]. These

Table 4: Average Pose accuracy across 6 datasets. Note that we use COLMAP poses of the edited frames as the "ground truth", and train out models on frames processed by COLMAP but without poses. The best results are highlighted in bold.

Metric	Nope-NeRF			Ours		
	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE$\downarrow$	RPE_t	RPE_r	ATE
GS25	0.080	0.038	0.006	**0.041**	**0.069**	**0.004**
Mip-NeRF	0.042	0.040	0.008	**0.043**	**0.035**	**0.004**
NeRFstudio	0.079	**0.034**	0.005	**0.039**	0.038	**0.003**
Co3D-V2	0.076	0.037	0.005	**0.037**	**0.067**	**0.003**
IN2N	**0.045**	0.076	**0.004**	0.046	**0.071**	0.004
Tanks & Temples	**0.046**	0.035	**0.004**	0.047	**0.033**	0.004

metrics provide a comprehensive assessment of the accuracy of estimated camera poses compared to ground truth, facilitating a direct comparison between different methodologies. The use of COLMAP poses of the edited frames as the "ground truth" ensures consistency in our evaluation protocol.

The results summarized in the Table 4 highlight the efficacy of our method across six datasets, demonstrating superior performance in terms of lower ATE and RPE for most datasets when compared to Nope-NeRF. This indicates that our approach not only accurately estimates the global trajectory of the camera but also maintains high local consistency in camera motion, underscoring the robustness and reliability of our pose estimation framework. The evaluation underscores our method's proficiency in accurately capturing the nuances of camera movement, reflecting its potential applicability in complex 3D scene reconstructions and AR/VR environments.

D User Study

To assess the subjective quality of scene editing, we orchestrated a user study that compared our approach with the leading state-of-the-art (SOTA) methods. Our user study probed into three critical attributes of the 3D-rendered images. First, we looked at **Text Fidelity**—the degree to which the rendered images conform to the provided text descriptions. Second, we considered **Con-**

Table 5: User Study. In a user study, 3DEgo attained top ratings in text fidelity and content preservation, whereas IN2N scored more favorably in terms of 3D scene consistency.

Method	Text Fidelity	Content Preservation	Scene Consistency
CSD [9]	5.7	5.9	6.1
DreamEditor [16]	6.3	6.0	6.2
ViCA-NeRF [5]	6.3	6.2	6.8
IN2N [6]	6.9	6.8	7.7
Ours	7.8	8.1	7.5

tent Preservation, focusing on our model's capability to modify the intended object with high accuracy. Lastly, we evaluated **Scene Consistency**, which measures the cohesiveness of the 3D scene's appearance from different viewing angles.

The detailed evaluation of user feedback presented in Table 5 demonstrates that our method outperformed in accurately mirroring text specifications and precisely modifying targeted objects. Users, numbering 150 and aged between 18 to 55, were invited to assess the quality of the edited scenes on a scale from 1 to 10, according to the three specified evaluation criteria. We report the mean score assigned to each method's output by the user in Table 5

E Autoregressive Editing with Diffusion Models

To enable consistent 2D editing across multiple views, our method integrates an autoregressive mechanism with the diffusion process. This approach ensures that the edited image is not only conditioned on the original image and textual prompts but also on adjacent edited images, enhancing consistency across views.

E.1 Autoregressive Conditioning

Incorporating concepts from Denoising Diffusion Probabilistic Models (DDPMs) [7] and autoregressive distribution [14], we propose a novel framework for consistent 2D editing across multiple views. This method leverages interpolated denoising and multi-view conditioning to ensure seamless integration of edits within a coherent scene representation DDPMs operate by gradually denoising a sample through a Markov chain process, transforming unstructured noise into structured data. Formally, this process is described as a series of conditional probabilities that model the reverse of adding Gaussian noise:

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}, \quad \text{with } p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{1}$$

where x_T is sampled from a normal distribution, and x_0 represents the target data distribution. The reverse process utilizes a learned noise prediction function $\varepsilon_\theta(x_t, t)$ to iteratively refine x_t towards x_0. Building upon the DDPM foundation, our framework introduces an autoregressive editing mechanism that conditions the editing of each frame on previously edited frames. This is in contrast to independent frame editing, which can lead to inconsistencies across views.

Given a sequence of frames $W = \{E_n\}_{n=1}^{w}$, we extend the noise prediction model to incorporate conditions from adjacent frames, effectively modeling an autoregressive distribution. This allows the edited frame to be influenced by its context within the sequence, enhancing consistency.

Interpolated denoising is employed to blend information across multiple views. For a target view, the noise prediction is calculated by a weighted sum of noise

estimates from adjacent views, ensuring that edits are coherent across the sequence. The weights may be determined by factors such as spatial proximity or temporal ordering, encapsulating the autoregressive nature of the editing process. By conditioning the denoising process on a set of views, we can ensure that the edited frame is consistent with both its immediate context and the broader set of views in the sequence. This is achieved through a combination of direct conditioning on available views and a learned interpolation of noise predictions across the view set.

F Broader Impact

Our project, **3DEgo**, introduces a transformative approach to reconstructing and editing 3D scenes from monocular videos, bypassing traditional Structure-from-Motion (SfM) and frame-by-frame editing techniques. This innovation makes 3D editing more accessible, reducing the need for specialized equipment and extensive computing power.

This research has practical implications across virtual and augmented reality, film production, and video game design, facilitating quick creation of virtual environments and immersive storytelling. Our auto-regressive editing technique also ensures consistency across multiple views, enhancing realism in VR and AR applications for improved user engagement in sectors like education and remote work.

We recognize the ethical challenges posed by easy editing capabilities, including concerns over content authenticity and misuse. It's crucial to establish ethical guidelines for usage, ensuring transparency and respect for privacy. Our goal is to continue advancing **3DEgo** while promoting discussions on its ethical use, aiming for a positive societal impact.

References

1. Bian, W., Wang, Z., Li, K., Bian, J.W., Prisacariu, V.A.: Nope-nerf: Optimising neural radiance field with no pose prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4160–4169 (2023) 4
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023) 1
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020) 1
4. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. arXiv preprint arXiv:2311.14521 (2023) 3
5. Dong, J., Wang, Y.X.: Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. Advances in Neural Information Processing Systems **36** (2024) 2, 5
6. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19740–19750 (2023) 5
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 6
8. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) **42**(4), 1–14 (2023) 1
9. Kim, S., Lee, K., Choi, J.S., Jeong, J., Sohn, K., Shin, J.: Collaborative score distillation for consistent visual editing. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), https://openreview.net/forum?id=0tEjORCGFD 2, 5
10. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023) 1
11. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: ICCV (2021) 4
12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 1
13. Rajič, F., Ke, L., Tai, Y.W., Tang, C.K., Danelljan, M., Yu, F.: Segment anything meets point tracking. arXiv preprint arXiv:2307.01197 (2023) 1
14. Uria, B., Côté, M.A., Gregor, K., Murray, I., Larochelle, H.: Neural autoregressive distribution estimation. Journal of Machine Learning Research **17**(205), 1–37 (2016) 6
15. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023) 3
16. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023) 5