
EA3D: Online Open-World 3D Object Extraction from Streaming Videos

Xiaoyu Zhou^{1†} Jingqi Wang^{1†} Yuang Jia¹ Yongtao Wang^{1*}
Deqing Sun² Ming-Hsuan Yang^{2,3}

¹Wangxuan Institute of Computer Technology, Peking University

²Google DeepMind ³University of California, Merced

Appendix

In this appendix, we provide additional content to complement the main paper:

- Appendix A: Datasets and Implementation Details.
- Appendix B: Method Details.
- Appendix C: Details of enhancing and comparing with our baseline methods.
- Appendix D: Novelty clarification against baselines.
- Appendix E: Model Efficiency Analysis.
- Appendix F: Detailed Ablation Studies.
- Appendix G: More Qualitative Visualizations.
- Appendix H: Supplementary Video.
- Appendix I: Diverse Downstream Applications.
- Appendix J: Failure Cases and Limitations.
- Appendix K: Broader Impacts.

A Datasets and Implementation Details

Datasets. We evaluate our method on two benchmarks. The LERF [8] dataset, captured with the iPhone Polycam app, features complex in-the-wild scenes. We use the extended version from [17], which includes ground-truth annotations for 3D object localization and 3D semantic segmentation. In addition, we manually annotated challenging open-vocabulary categories and hard cases to enable a more comprehensive evaluation of our method. The ScanNet [3] dataset comprises a diverse set of indoor scenes with a rich variety of objects. While it offers RGB-D images and 3D meshes, our pipeline utilizes only the RGB image sequences. Consistent with prior work such as EmbodiedSAM [25], we use the same high-quality indoor scenes and labeled point clouds for evaluation. For semantic evaluation, we compute metrics using all ground truth classes from LERF and ScanNet. Categories predicted by the VLMs may be absent from the ground truth due to the benchmark’s limited semantic classes. To ensure consistency with baseline methods, we prompt VLMs to merge such categories with the closest predefined classes—for example, combining “bookshelf” and “bookcase” under “bookcase.”

Implementation Details. We implement EA3D on top of HiCoM, with reduced training iterations to ensure rapid Gaussian updates. Each incoming frame undergoes 100 update steps, followed by another 100 training steps after the incorporation of new Gaussians. The Gaussian parameters are initially initialized based on the estimated odometry and corresponding camera poses. Low-opacity

*Corresponding author.

Gaussians are removed prior to training on the next frame, allowing them to still contribute to the current representation. We employ the off-the-shelf CogVLM [22, 6] model to interpret the scene. For semantic feature map extraction, we utilize the Grounded-SAM and CLIP models, with ViT-Huge serving as the image encoder. To enhance efficiency, we apply a 2× downsampling to the input image before feeding it into the feature extractor. At the time of this work, the official code released by baseline methods exhibited instability and execution issues. Therefore, we report experimental results based on our own implementation. All experiments are conducted using PyTorch on a single 80GB A100 GPU.

B Method Details

Open-world interpretation by VLMs. We present a detailed illustration of how open-world scene understanding is obtained online from VLMs, as shown in Fig. II. We use the key prompt “find, identify, and analyze anything in the scene” to guide VLMs in extracting object categories from single-frame images, which are then dynamically updated into an online semantic cache. Notably, the semantics extracted by VLMs may contain ambiguities or redundancies. We address semantic ambiguities in the Method section of the main text. To reduce redundancy, we adopt a semantic fusion strategy that avoids repeatedly storing similar or overlapping concepts in the cache. Specifically, each semantic label is encoded into a feature vector $T \in \mathbb{R}^{1 \times V}$ using a pretrained text encoder from CLIP [31, 29]. We compute pairwise similarities between these vectors and merge those exceeding a predefined similarity threshold ϑ . For example, “brown toy bear” and “brown teddy bear” are merged, while semantically distinct concepts like “chair” and “sofa” remain separate. More ablation about the semantic cache updating threshold ϑ is further conducted in Section F. For semantic cache updating threshold, we first employ an aggregation strategy for physical attributes via instance-level feature map fusion, performed during the online cache update. In this process, physical attribute features with the highest occurrence frequency and confidence are dynamically fused into the online semantic cache as a variable-length vector, under the constraint of multi-view 3D consistency.

Online Gaussian Splatting. 3D Gaussian Splatting explicitly represents scenes using anisotropic 3D Gaussian primitives, including position μ , covariance matrix Σ , opacity o , and spherical harmonics coefficients (SH):

$$G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}. \quad (1)$$

The covariance matrix Σ is decomposed into a scaling matrix \mathbf{S} and a rotation matrix \mathbf{R} to ensure physical meaning and facilitate optimization:

$$\Sigma = \mathbf{R} \mathbf{S} \mathbf{S}^T \mathbf{R}^T, \quad (2)$$

where $\mathbf{S} = \text{diag}(s_x, s_y, s_z) \in \mathbb{R}^3$ and $\mathbf{R} \in SO(3)$ are parameterized by a 3D scaling vector \mathbf{s} and a rotation quaternion \mathbf{q} , respectively. Each Gaussian primitive is further enriched with color and opacity, represented by spherical harmonic coefficients \mathbf{h} and a scalar α , respectively. We further augment GS with fused features from VLMs and VFMs, comprising semantic features \mathbf{S} , physical attribute features \mathbf{Y} , and a continuous vector $T \in \mathbb{R}^{1 \times V}$ retrieved from an online semantic cache Ω . To render a novel viewpoint, Gaussian primitives are projected onto the camera plane with alpha-blending to accumulate the final splatted feature \hat{F} :

$$\hat{F} = \sum_{i \in N} F_i \cdot \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

where α_i denotes the opacity, F_i is the integrated feature map of the i -th Gaussian. The contributions of N overlapping Gaussian primitives at each pixel account for their depth-ordering.

Gaussian2Voxel Splatting. Inspired by [34], we use accumulated Gaussians to splat onto the voxel grid at an arbitrary voxel size to generate the occupancy, with each voxel’s occupancy determined by weighting the occupied range and opacity of the Gaussians:

$$F(o) = \sum_{i=1}^N d_i G(x_i) \alpha_i \text{softmax}(\mathbf{F}_i), \quad (4)$$

where d_i is the occupied depth of the Gaussian2voxel, treated as the splatting weight coefficient. α_i is the opacity, \mathbf{F}_i is the integrated feature map.

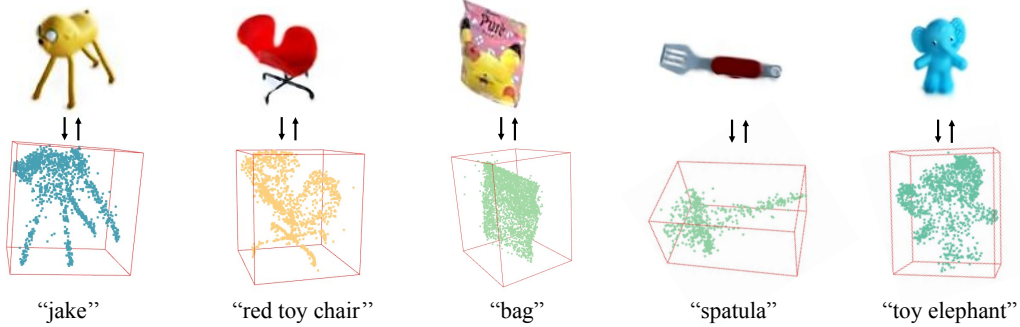


Figure I: **Visualization of Semantic-aware splatting to 3D Bbox and Semantic Occupancy.**

3D Bbox Estimation. For each online feature Gaussian, we generate category-specific boundaries by applying a KNN clustering algorithm to select the boundary ranges of Gaussian ellipsoids sharing the same semantic category. The spatial coordinates of semantic cluster centers serve as the 3D bounding box centers. The bounding box dimensions (i.e., length, width, and height) are determined by applying the Axis-Aligned Bounding Box (AABB) algorithm to enclose the Gaussian ellipsoids within the cluster, based on their intersections with the bounding box edges.

3D Mesh Generation. Following PGSR [2], we start surface extraction by rendering the depth from our online feature Gaussians for each training view. We then apply the TSDF Fusion [15, 23] algorithm to construct the corresponding TSDF field, from which the mesh is subsequently extracted.

C Details of Enhancing and Comparing with Baseline Methods

Since we are the first to propose online 3D object extraction without relying on 3D geometric priors, poses and predefined category lists, we enhance prior and concurrent methods to serve as stronger baselines. All enhanced baselines are reimplemented and refactored from their original codebases. Detailed implementation will be made available upon acceptance of the paper.

Streaming Gaussian + Open-Vocabulary Segmentation. Although streaming Gaussian-based methods [20, 5, 26] enable scene reconstruction from video streams, they suffer from critical limitations, including the need for initial multi-view coverage and pre-known camera poses. Moreover, their inability to understand the scene semantics makes them unsuitable for the 3D object extraction task. To address this issue, we enhance our main baseline HiCOM [5] by integrating online streaming Gaussian optimization with VFM-guided semantics. Specifically, HiCOM incrementally reconstructs the scene Gaussians frame by frame, while each frame’s 2D semantic segmentation—generated by VFMs—is lifted and projected onto the reconstructed Gaussians through a two-stage 2D-to-3D mapping process. As shown in Tab. I, our method outperforms the enhanced baseline, achieving a better trade-off between accuracy and speed. In contrast, the compared baselines exhibit noticeable quality degradation due to the lack of joint optimization for scene geometry and understanding.

Online SLAM + Open-Vocabulary Segmentation. SLAM-based methods allow for online mapping of scenes with unknown poses but are highly dependent on accurate geometric priors from depth input and expensive post-refinement. They also fail to simultaneously reconstruct geometry and understand scenes. To overcome this challenge, we integrate VFM-guided semantics into SLAM-based online mapping systems, synchronously projecting the acquired semantic priors onto the constructed point cloud or 3D Gaussians. For a fair comparison, the baseline excludes the post-refinements, which are typically considered offline procedures. As shown in Table I, SLAM-based methods struggle to jointly recover accurate geometry and semantics without costly post-processing and face ambiguity in complex scenes [8].

Feature-distillation Gaussian + Streaming Update. Feature Gaussians [18, 33, 32] propose to distill object-centric vision-language features into 3D Gaussians within the same optimization pipeline as vanilla 3DGS. However, these methods operate offline and cannot support online 3D object extraction. We introduce an online Gaussian update [5] combined with feature distillation [18] to achieve semantic-aware online Gaussians. Notably, since the feature-carrying 3D Gaussians proposed by [18, 32] do not support incremental online updates, we first use HiCOM to add new Gaussians

Table I: Comparisons on ScanNet [3]. The best results are highlighted in **bold**, and the second-best results are underlined. “*” indicates the use of the colmap-estimated poses following [28, 17, 18]. “—” indicates that the method does not support the specified task. “Rec., Seg., Bbox., Occ.” denotes four multi-task evaluations: reconstruction quality, instance segmentation, 3D bounding box estimation, and semantic occupancy estimation. “Speed” refers to the training speed, measured in frames per second (FPS).

Task:				Rec.		Seg.		Bbox.		Occ.	
Method	Input	Online	Speed	PSNR	SSIM	mIoU	mAcc	AP	mAP	IoU	mIoU
HiCOM [5]	RGB	✓	0.29	22.6	0.82	34.8	61.9	✗	✗	✗	✗
HiCOM [5]+VFM [19]	RGB	✓	0.11	22.6	0.82	34.8	61.9	52.5	23.8	42.4	27.9
MonoGS [14]	RGBD	✓	0.18	24.3	0.85	36.3	60.5	✗	✗	✗	✗
MonoGS [14]+VFM [19]	RGBD	✓	0.07	24.3	0.85	36.3	60.5	51.7	27.7	44.5	27.2
SGS-slam [11]+VFM [19]	RGBD	✓	0.05	20.7	0.78	33.5	57.8	45.6	25.2	35.4	22.0
FeatureGS [18]	RGB	✗	0.01	23.9	0.84	41.1	66.0	51.4	32.7	50.9	31.2
FeatureGS [18]+HiCOM [14]	RGB	✓	0.03	24.5	0.85	40.8	66.3	55.8	34.7	50.7	31.4
Feat-3dgs [32]+HiCOM [14]	RGB	✓	0.02	23.3	0.84	38.9	63.5	50.1	28.6	49.2	30.5
LSM [4]	RGB	✓	0.89	24.3	0.80	40.2	61.7	✗	✗	✗	✗
SAM3D [27]	Points	✓	0.92	✗	✗	39.2	62.3	53.7	29.1	53.3	26.7
Cut3R [21]+SAM3D [27]	RGB	✓	0.41	✗	✗	40.3	62.5	50.6	26.4	46.6	25.3
EA3D*	RGB	✓	0.20	<u>25.5</u>	<u>0.87</u>	<u>45.9</u>	<u>71.2</u>	59.2	<u>39.6</u>	55.0	34.3
EA3D	RGB	✓	0.23	25.8	0.89	46.3	71.8	57.9	39.9	55.4	<u>33.9</u>

per frame, then distill features into them sequentially. While this two-stage process largely preserves the effectiveness of the original method, it introduces significant runtime disruptions. As shown in Table I, our method outperforms this strong baseline by enabling end-to-end online updating and joint optimization of feature-rich Gaussians, enhancing performance while maintaining model efficiency.

Cut3R + SAM3D. Cut3R [21] enables the online generation of metric-scale point maps (per-pixel 3D points) from video streams. However, Cut3r struggles to preserve fine geometric details, photorealistic rendering, and scene understanding. It can also generate extremely blurry and distorted results when extrapolating far from observed views. We enhance Cut3r with scene understanding by integrating SAM3D [27], employing a bidirectional merging strategy to project 2D masks into 3D. Results in Table I show that our method outperforms the extended Cut3r, delivering higher-quality geometry and rendering without a significant increase in computational cost. Moreover, Cut3r generates a large number of redundant per-pixel 3D points, which interfere with semantic projection. In contrast, our method employs an online Gaussian update strategy to remove redundant Gaussians while implicitly aligning semantics in 3D space.

D Novelty Clarification Against Baselines

Here, we further clarify the distinctions and advantages of our proposed method compared to concurrent works.

Compared with Feature Gaussian methods. Current feature Gaussian splatting methods [18, 32, 33, 9] aim to equip GS with scene understanding via feature field distillation but remain tied to the fully offline vanilla 3DGS pipeline. While these approaches combine semantic feature gradients with Gaussian attribute updates, they lack an explicit joint optimization strategy for geometry and semantics, often resulting in slow convergence. Notably, the distilled features in these methods are predefined, with fixed semantic categories that remain unchanged throughout the optimization process. In contrast, our method adopts a fully online feature embedding strategy with a simple yet effective feedforward update mechanism, enabling dynamic and adaptive feature refinement, which enhances the pipeline’s generalization.

Compared with Streaming Gaussian methods. Current streaming Gaussian methods [20, 5, 26] face three challenges: 1) they require multi-view video streams, which incur high capture costs in practical applications; 2) they depend on pre-known camera poses; 3) they cannot jointly optimize scene geometry and understanding in a synchronized manner. In contrast, building on streaming

User


Hi, please find, extract, and analyze all the objects you can see in the scene. For each object, specify its semantic category and physical attributes, and compile a scene description list formatted as {ID, semantic category, physical attributes}.

1, Wooden table, Wood

2, Cookies bag, Kraft paper

3, Plate, Smooth white ceramic

4, Cookies, Chocolate chips



User

Hi, please find, extract, and analyze all the objects you can see in the scene. For each object, specify its semantic category and physical attributes, and compile a scene description list formatted as {ID, semantic category, physical attributes}.

1, Wooden table, Wood

3, White plate, Smooth white ceramic


4, Mug with Hot Drink, White ceramic

5, Spoon, Stainless steel

6, Coffee Capsule (Empty), Plastic

7, Liquid Spill, Coffee or syrup

8, Plush Object, Synthetic fabric



VLM

Online Semantic Cache		
ID	Semantic Category	Physical Attributes
1	Wooden table	Wood
2	Cookies bag	Kraft paper
3	White plate	Ceramic
4	Cookies	Chocolate chips
...
6	Mug	Ceramic
7	Spoon	Stainless steel
8	Coffee Capsule	Plastic
9	Liquid Spill	Coffee or syrup
generating...	generating...	generating...

Figure II: Visualization of Open-world interpretation by VLMs.

Gaussians, we introduce an online visual odometry that enables incremental reconstruction from monocular dynamic video streams. Additionally, we design a knowledge-fusion streaming feature update strategy to ensure rapid optimization of both geometry and scene understanding.

Compared with 2D-to-3D Lifting methods. A straightforward way to obtain 3D scene understanding from 2D foundational models is to lift the 2D results into 3D using a voting fusion algorithm combined with 2D-to-3D projection. Previous approaches [27, 30, 16, 7, 1] leverage SAM to segment 2D images and project the results onto pre-constructed 3D representations such as point clouds, meshes, or 3DGS. Our method differs from these approaches in two key aspects: 1) EA3D does not rely on prebuilt 3D representations but simultaneously construct scene geometry and semantics; 2) EA3D achieves efficient 3D spatial alignment through hybrid feature embeddings rather than directly projecting decoded 2D outputs. Experimental results demonstrate that our approach outperforms lifting methods, effectively addressing semantic ambiguity and occlusion issues inherent in 2D-to-3D.

Compared with online SLAM methods. SLAM-based methods online 3D scene mapping without known camera poses. Recent advances [11, 10, 36, 35] extend SLAM to scene understanding by incorporating semantic information to provide additional supervision for semantic scene mapping. However, these methods require additional depth ground truth as input, which is difficult to obtain in real-world applications. They also rely on 2D semantic segmentation masks and costly post-processing for global semantic bundle adjustment. In contrast, EA3D is a fully end-to-end online 3D object extraction method that requires no geometric priors or costly post-processing. Our method offers greater flexibility, supporting open-world 3D semantic understanding and multi-level geometric construction. EA3D also outperforms these SLAM-based methods in training and rendering speed, reconstruction quality, and semantic accuracy.

E Model Efficiency Analysis

A comprehensive comparison of model efficiency is shown in Tab. II, including module-wise breakdown, training time, rendering speed and quality, model size, and memory usage. EA3D utilizes joint online visual odometry and Gaussian optimization, both of which are faster than offline approaches. Despite leveraging additional visual base models to enhance the understanding of long-tail objects in the open world, our method maintains a comparable or even faster feature embedding speed as we extract image features without the need for a decoding process. In contrast, LangSplat [17] and GSGrouping [28] require feature decoding during the training phase, which is time-consuming. Our method strikes a balance between speed and accuracy, ensuring higher rendering efficiency and lower storage overhead.

Table II: **Efficiency and Performance Comparison.** Since all baseline methods perform offline reconstruction, we report the average runtime per component and total pipeline by measuring the execution time across all training views of the entire scene for them. “Total” indicates the average training speed per frame, “Render” refers to the rendering speed, and “Parameters” represent all trainable parameters in the pipeline.

Method	Component	Online	Component (FPS \uparrow)	Total (FPS \uparrow)	Quality (PSNR \uparrow)	Render (FPS \uparrow)	Parameters (M \downarrow)
LERF [8]	Colmap	✗	0.49	0.03	16.5	67	1272
	Whole Seg.	✗	0.12				
	GS Training	✗	0.06				
LangSplat [17]	Colmap	✗	0.49	0.08	18.4	140	714
	Whole Seg.	✗	0.13				
	GS Training	✗	0.26				
GSGrouping [28]	Colmap	✗	0.49	0.06	19.6	180	460
	Whole Seg.	✗	0.19				
	GS Training	✗	0.13				
OpenGaussian [24]	Colmap	✗	0.49	0.05	22.1	120	528
	Whole Seg.	✗	0.14				
	GS Training	✗	0.10				
FeatureGS [18]	Colmap	✗	0.49	0.07	23.9	190	647
	Whole Seg.	✗	0.23				
	GS Training	✗	0.15				
Ours	Online Odo	✓	1.67	0.23	25.8	210	364
	Feature Embed.	✓	0.43				
	GS Training	✓	0.84				

F Detailed Ablation Studies

Sensitivity to the online semantic cache. The online semantic cache dynamically updates the extracted object categories as new observations arrive. We conduct ablation studies to evaluate the effectiveness of the dynamic updating semantic cache and perform sensitivity analysis on the updating threshold ϑ . As shown in Tab. V, the online semantic cache facilitates more comprehensive extraction of open-world semantics from the scene, while also enabling more effective query-based retrieval of semantic features. Our method also demonstrates strong robustness across different updating thresholds, where ambiguous semantics are implicitly corrected during the multi-view reconstruction and understanding.

Importance of multi-level knowledge feature fusion. The integrated knowledge features contribute to a more comprehensive understanding of multi-level semantics in the scene by fusing representations from Grounded-SAM and CLIP. We validate the effectiveness of this module by ablating each feature extractor individually. Results reveal that relying on a single visual foundation model often introduces ambiguity: CLIP features overemphasize high-frequency regions, hindering accurate instance localization, while DINO and SAM focus on low-frequency structures, often missing fine-grained object details. Ablation results indicate that multi-level feature fusion contributes to more comprehensive and finer semantic feature extraction.

Importance of dense online visual odometry. Online visual odometry facilitates the generation of high-quality initial poses and relatively dense point cloud priors. We validate the effectiveness of the online visual odometry by replacing the dense odometry point cloud used in our method with a sparse point cloud estimated from SfM (Colmap), a commonly used offline pose estimator in traditional 3D reconstruction and understanding pipelines. Experimental results indicate that a sparse initial point cloud from SfM results in unreliable keypoint matching and struggles to capture fine-scale structures or small objects. In contrast, our method leverages a fused dense point cloud from Cut3R [21], enabling more accurate and timely reconstruction of detailed geometry.

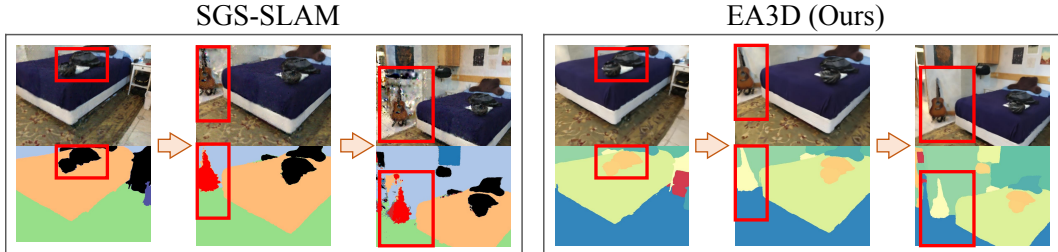


Figure III: Compare EA3D with traditional SLAM-based methods.

Table III: Robustness evaluation under challenging conditions, including severe occlusion, rapid camera motion, and low-texture environments. “Rec.” reflects the accuracy of online geometry and visual odometry, while “Seg.” represents multi-view 3D understanding, which can be used to assess semantic coherence.

Methods	Occlusion	Fast motion	Low-texture
Baseline (Rec.)	18.4	20.2	22.3
Ours (Rec.)	23.1	23.3	22.9
Baseline (Seg.)	31.6	34.8	35.8
Ours (Seg.)	39.5	41.1	44.3

Design of online feature Gaussian. To enable online streaming scene reconstruction and understanding, we propose a strategy based on online Gaussian feature optimization. Existing online reconstruction methods can be roughly categorized into two types: 1) StreamGS-based approaches [20, 5, 13], which require multi-view video streams and pre-defined camera poses, and 2) Online SLAM-based methods [12, 14], which struggle with modeling dynamic movements and fine-grained geometry, and rely heavily on expensive post-refinement for satisfactory reconstruction and rendering quality. More critically, both types of methods lack scene understanding capabilities and are unable to capture object-level semantics and geometry in an online manner. Inspired by both paradigms, we propose an online framework that enables real-time reconstruction of large-scale environments and fine-grained object geometry, while simultaneously inferring semantic information. EA3D integrates SLAM-based online pose estimation and further enhances the reconstruction of complex objects via dense, per-pixel geometry modeling and feature embedding. In contrast to HiCoM [5], our baseline method, EA3D operates without relying on external pose priors or multi-view streaming input, enabling plug-and-play scene reconstruction in dynamic environments and offering greater applicability in real-world scenarios.

Effectiveness of regularization term. We further ablate the effect of semantic-awareness regularization term \mathcal{L}_s by removing it. Ablation shows that regularization term facilitates both geometric reconstruction and rendering quality, which helps optimize instance-level Gaussian distributions, thereby better promoting the joint optimization of semantic knowledge and scene geometry.

Robustness under challenging conditions. To further quantify the robustness and accuracy of our model under various challenging conditions, we thoroughly collected scenes and video clips from the benchmark that feature severe occlusion, rapid camera motion, and simulated low-texture environments. Targeted validation experiments were conducted on these challenging cases, as presented in the Table III. Our method demonstrates outstanding robustness and accuracy, surpassing the baseline approaches even under such difficult conditions.

Hyperparameters. We provide a further ablation study on the hyperparameters in our method, including the loss weight balancing factors λ_1 , λ_2 , and λ_3 , odometry update threshold ϱ , pruning threshold ζ , and the semantic cache updating threshold ϑ . As shown in Tab. IV and Tab. V, our method demonstrates strong robustness to hyperparameter variations.

G More Qualitative Visualizations

We provide additional qualitative visual comparisons as shown in Fig. IV and Fig. III. Results demonstrate that our online feature Gaussians capture both geometric structure and semantic context

Table IV: Ablation on hyperparameters λ_1 , λ_2 , and λ_3 .

λ_1	λ_2	λ_3	Rec.(PSNR \uparrow)	Seg.(mIoU \uparrow)
0.10	0.25	0.10	25.7	45.9
0.15	0.15	0.20	25.3	46.3
0.20	0.20	0.20	25.6	46.0
0.25	0.10	0.15	25.8	46.3

Table V: Ablation on hyperparameters, including odometry update threshold ϱ , pruning threshold ζ , and the semantic cache updating threshold ϑ .

ϱ	PSNR \uparrow	mIoU \uparrow	ζ	PSNR \uparrow	mIoU \uparrow	ϑ	PSNR \uparrow	mIoU \uparrow
0.5	24.9	45.8	2×10^{-2}	24.8	45.6	0.5	25.4	45.9
0.6	25.4	46.1	2×10^{-3}	25.5	45.9	0.6	25.8	46.3
0.7	25.8	46.3	2×10^{-4}	25.8	46.3	0.7	25.7	45.4
0.8	25.6	46.0	2×10^{-5}	25.0	46.1	0.8	25.8	45.0

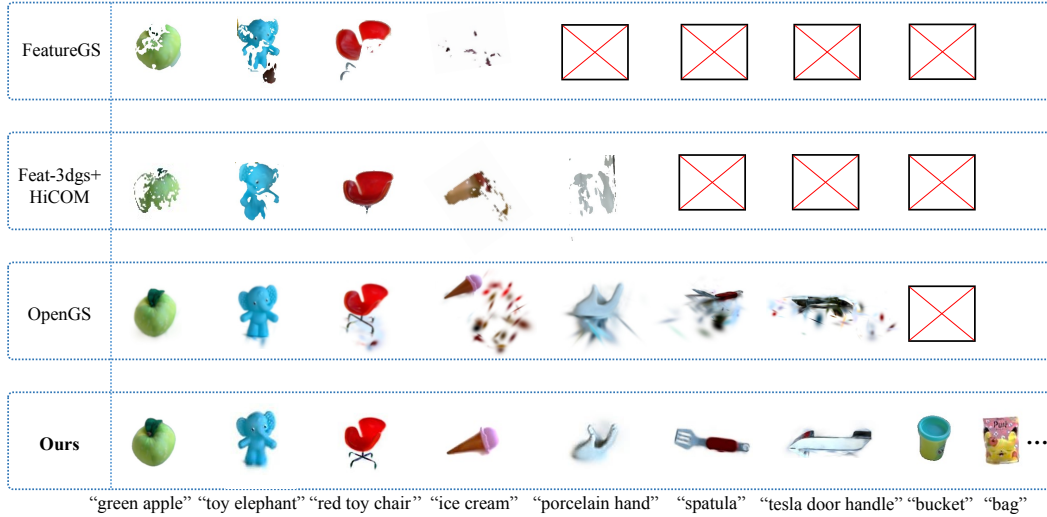


Figure IV: **Visualization of 3D Object Extraction.**

with remarkable efficiency and precision. By jointly optimizing for pose estimation and feature representation, our method produces coherent, high-fidelity reconstructions that preserve fine details and semantic consistency. In contrast, state-of-the-art baselines often suffer from noisy feature aggregation, leading to degraded rendering quality and a failure to recognize or reconstruct complex or ambiguous objects—particularly those with limited observations or underrepresented categories.

H Supplementary Video

Please refer to our supplementary video for a more comprehensive visualization of online 3D object extraction.

I Diverse Downstream Applications

EA3D facilitates diverse downstream applications by dynamically aligning with LLM instructions or text-to-image generation models. As illustrated in Fig. V, combining EA3D with controllable generation and editing enables compelling functionalities such as manipulation simulation, motion emulation, controllable 3D editing, and object insertion or removal.

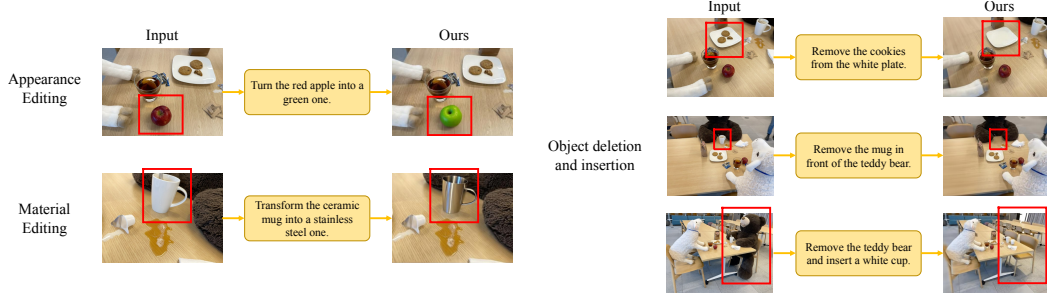


Figure V: Visualization of Diverse Downstream Applications.

J Failure Cases and Limitations

The primary limitation of our approach arises from the imperfect accuracy and completeness of semantic extraction by vision-language models (VLMs) and vision foundation models (VFM) in open-world scenarios. In particular, when VLMs generate incorrect semantic interpretations, our method may struggle to fully rectify these errors, leading to semantic mismatches within certain geometric regions. Although our approach supports implicit semantic alignment and correction, it fails to reconstruct geometry or resolve semantic ambiguity for objects that appear in only a few frames (e.g., a single frame in the entire video). This limitation is inherent to the underlying principles of multi-view reconstruction. In future work, we plan to integrate autoregressive and diffusion-based generative models to enable robust geometric and semantic reasoning under single-view or severely occluded conditions.

K Broader Impacts

This paper presents research aimed at advancing the fields of 3D vision, which hold significant promise for enhancing the 3D object extraction. While AI-driven scene reconstruction and perception bring benefits, they could also raise concerns regarding their social and economic impacts. Automating 3D labeling and perception tasks can potentially disrupt the labor market, posing risks to certain job sectors, particularly in sectors that rely on manual data annotation. It is crucial to exercise caution and ensure that the societal implications are thoroughly addressed.

References

- [1] Rohan Chacko, Nicolai Haeni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by gaussians: A simple, fast and flexible method for 3d instance segmentation. *arXiv preprint arXiv:2502.00173*, 2025. 5
- [2] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3
- [3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1, 4
- [4] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *NIPS*, 37:40212–40229, 2024. 4
- [5] Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. Hicom: Hierarchical coherent motion for streamable dynamic scene with 3d gaussian splatting. *arXiv preprint arXiv:2411.07541*, 2024. 3, 4, 7
- [6] Wenyi Hong, Weihaan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 2
- [7] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. In *ECCV*, 2024. 5

- [8] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lrf: Language embedded radiance fields. In *ICCV*, pages 19729–19739, 2023. 1, 3, 6
- [9] Hao Li, Roy Qin, Zhengyu Zou, Diqi He, Bohan Li, Bingquan Dai, Dingwen Zhang, and Junwei Han. Lang-surf: Language-embedded surface gaussians for 3d scene understanding. *arXiv preprint arXiv:2412.17635*, 2024. 4
- [10] Kunyi Li, Michael Niemeyer, Nassir Navab, and Federico Tombari. Dns-slam: Dense neural semantic-informed slam. In *IROS*, pages 7839–7846. IEEE, 2024. 5
- [11] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *ECCV*, 2024. 4, 5
- [12] Renwu Li, Wenjing Ke, Dong Li, Lu Tian, and Emad Barsoum. Monogs++: Fast and accurate monocular rgb gaussian slam. *arXiv preprint arXiv:2504.02437*, 2025. 7
- [13] Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-hong Kao, Ying-Cong Chen, and Yan Lu. Streamgs: Online generalizable gaussian splatting reconstruction for unposed image streams. *arXiv preprint arXiv:2503.06235*, 2025. 7
- [14] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *CVPR*, pages 18039–18048, 2024. 4, 7
- [15] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 3
- [16] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *CVPR*, 2024. 5
- [17] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, pages 20051–20060, 2024. 1, 4, 5, 6
- [18] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature splatting: Language-driven physics-based scene synthesis and editing. *arXiv preprint arXiv:2404.01223*, 2024. 3, 4, 6
- [19] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 4
- [20] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *CVPR*, pages 20675–20685, 2024. 3, 4, 7
- [21] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 4, 6
- [22] Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *NeurIPS*, pages 121475–121499, 2024. 2
- [23] Jiaxin Wei and Stefan Leutenegger. Gsfusion: Online rgb-d mapping where gaussian splatting meets tsdf fusion. *IEEE Robotics and Automation Letters*, 2024. 3
- [24] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *arXiv preprint arXiv:2406.02058*, 2024. 6
- [25] Xiuwei Xu, Huangxing Chen, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Embodiedsam: Online segment any 3d in real time. *arXiv preprint arXiv:2408.11811*, 2024. 1
- [26] Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. *arXiv preprint arXiv:2503.16979*, 2025. 3, 4
- [27] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 4, 5

- [28] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, pages 162–179, 2024. [4](#), [5](#), [6](#)
- [29] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *CVPR*, 2023. [2](#)
- [30] Dingyuan Zhang, Dingkan Liang, Hongcheng Yang, Zhikang Zou, Xiaoqing Ye, Zhe Liu, and Xiang Bai. Sam3d: Zero-shot 3d object detection via segment anything model. *arXiv preprint arXiv:2306.02245*, 2023. [5](#)
- [31] Kaiyang Zhou, Jingkan Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. [2](#)
- [32] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, 2024. [3](#), [4](#)
- [33] Shijie Zhou, Hui Ren, Yijia Weng, Shuwang Zhang, Zhen Wang, Dejia Xu, Zhiwen Fan, Suyu You, Zhangyang Wang, Leonidas Guibas, et al. Feature4x: Bridging any monocular video to 4d agentic ai with versatile gaussian feature fields. *arXiv preprint arXiv:2503.20776*, 2025. [3](#), [4](#)
- [34] Xiaoyu Zhou, Jingqi Wang, Yongtao Wang, Yufei Wei, Nan Dong, and Ming-Hsuan Yang. Autoocc: Automatic open-ended semantic occupancy annotation via vision-language guided gaussian splatting. *arXiv preprint arXiv:2502.04981*, 2025. [2](#)
- [35] Siting Zhu, Renjie Qin, Guangming Wang, Jiuming Liu, and Hesheng Wang. Semgauss-slam: Dense semantic gaussian splatting slam. *arXiv preprint arXiv:2403.07494*, 2024. [5](#)
- [36] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *CVPR*, pages 21167–21177, 2024. [5](#)

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We claim the main contribution of this paper in both the Abstract and Introduction sections.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitation of this work in the Supplementary materials.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide the implementation details in Section ??.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not provide new datasets and will release partial code after the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the training details and hyperparameters in Section ??.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported because it would be too computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the information for computer resources in Section ??.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research in the paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide the discussion of broader impacts in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The models in this paper pose no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All owners of models, code, and data we used are properly cited. We compliance all licenses of models, code, and data.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.