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

Xiaoyu Zhou[1†] **Jingqi Wang**[1†] **Yuang Jia**[1] **Yongtao Wang**[1*]
Deqing Sun[2] **Ming-Hsuan Yang**[2, 3]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Google DeepMind [3]University of California, Merced

Figure 1: Illustration of ***EA3D***, which enables online open-world 3D object extraction. Given a streaming video as input with unknown geometry, pose, or semantics, EA3D performs online and simultaneous scene interpretation and geometry reconstruction, enabling multi-task understanding and modeling of any 3D objects in the scene.

Abstract

Current 3D scene understanding methods are limited by offline-collected multi-view data or pre-constructed 3D geometry. In this paper, we present ***ExtractAnything3D*** (EA3D), a unified online framework for open-world 3D object extraction that enables simultaneous geometric reconstruction and holistic scene understanding. Given a streaming video, EA3D dynamically interprets each frame using vision-language and 2D vision foundation encoders to extract object-level knowledge. This knowledge is integrated and embedded into a Gaussian feature map via a feed-forward online update strategy. We then iteratively estimate visual odometry from historical frames and incrementally update online Gaussian features with new observations. A recurrent joint optimization module directs the model's attention to regions of interest, simultaneously enhancing both geometric reconstruction and semantic understanding. Extensive experiments across diverse benchmarks

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-1833

and tasks, including photo-realistic rendering, semantic and instance segmentation, 3D bounding box and semantic occupancy estimation, and 3D mesh generation, demonstrate the effectiveness of EA3D. Our method establishes a unified and efficient framework for joint online 3D reconstruction and holistic scene understanding, enabling a broad range of downstream tasks. The project webpage is available at https://github.com/VDIGPKU/EA3D.

1 Introduction

To see is, as famously defined by David Marr [25], "*to know what is where by looking.*" For an autonomous agent, such as a robot, operating in an unfamiliar environment, this translates into formidable challenges. Imagine a robot entering a new room, observing and understanding its surroundings on the fly (Fig. 1). It faces an unknown quantity and variety of objects (**open world**) and needs to process unfamiliar 3D geometry (**unknown geometry**) in a streaming mode (**online exploration**). To effectively navigate and interact within such a dynamic 3D space, the robot must be able to dynamically construct open-world 3D representations of the scene. Concurrently, it must comprehend the geometric structures and physical properties of the objects it encounters and perceptively model the motion states of all semantic entities within complex, evolving environments.

While Vision-Language Models (VLMs) [13, 57, 21] show impressive results on 2D open-world understanding , they struggle in 3D domains, exhibiting view inconsistencies[58, 11], geometric misalignment[1], and inability to handle occlusions. A straightforward solution is to lift 2D VLM outputs into 3D using scene geometry [49, 56, 14], but this requires pre-constructed 3D geometry, annotated datasets for training, and still suffers from 3D-2D misalignment issues. Recent differentiable rendering frameworks like NeRF [27, 36] and 3DGS [15, 54, 8] enable joint 3D scene understanding by optimizing 3D representations with pixel-level pseudo-labels[16, 52, 65, 31]. However, these offline approaches require complete multi-view images and time-consuming multi-stage processes.

In this paper, we introduce ***ExtractAnything3D*** (EA3D), an online open-world scene understanding framework that simultaneously explores, reconstructs, and interprets the 3D geometry and semantic knowledge of a scene. Similarly to human perception, our system starts processing streaming visual inputs as soon as it enters a room, reconstructing and understanding the current scene online based on historical observations and prior knowledge. As new frames emerge, they progressively reveal more comprehensive spatial information, enriching the internal knowledge base and allowing the system to infer occluded regions via novel view synthesis.

Specifically, we utilize VLMs to openly interpret object categories and physical properties from the emerging frame while dynamically maintaining a semantic cache. We then combine features from multiple visual foundation models with semantic cues to construct a dynamically updated knowledge-integrated feature map. The knowledge-integrated features are embedded into Gaussian representations through a fast feedforward step and are updated jointly over time. To incrementally extract both geometry and knowledge of 3D objects in an online manner, we construct Online Feature Gaussians, consisting of two core components: online visual odometry and online Gaussian updating. Benefiting from a recurrent joint optimization strategy, our proposed Online Feature Gaussians dynamically extract any 3D objects in the scene, facilitating multiple tasks including photo-realistic rendering, semantic and instance segmentation, physical property analysis, and geometric reasoning (e.g., 3D bounding boxes, semantic occupancy, and 3D mesh generation). EA3D thus establishes a unified and efficient framework for joint online 3D reconstruction and holistic scene understanding, enabling a wide range of downstream tasks.

The contributions of this work are: 1) We propose a unified online open-world 3D objects extraction framework enabling simultaneous online reconstruction and understanding without geometric or pose priors. 2) Taking streaming video as input, our method effectively leverages historical knowledge to guide 3D object extraction at the current observation, enabling online joint updates of integrated features and delivering high-quality, efficient geometric reconstruction and scene understanding. 3) Our method supports a broad set of tasks, including photo-realistic reconstruction and rendering, semantic and instance segmentation, 3D bounding box construction, semantic occupancy estimation, and 3D mesh generation, consistently achieving good performance across multiple benchmarks.

https://doi.org/10.52202/085713-1833

2 Related Work

Open-World Foundation Model. When exploring the real world, the quantity and categories of 3D objects remain unknown in unbounded environments. Recent advances in Vision-Language Models (VLMs) and Vision Foundation Models (VFMs) have significantly advanced open-world interpretation of 2D images. VLMs [13, 21, 57, 42] effectively fuse visual and textual cues for Visual Question Answering (VQA), while SAM-based [17, 33] and CLIP-based methods [9, 22, 51] excel in generalized semantic segmentation and instance detection. However, these methods suffer from severe multi-view inconsistencies and semantic ambiguities, especially for small objects, due to their limited geometric awareness. They also struggle with spatial occlusions and suffer from memory degradation over time. To overcome these challenges, we propose an online, synchronized framework for joint reconstruction and understanding, where 2D foundational features are implicitly aligned throughout the online reconstruction process. Our framework leverages online embedding from VFMs and recurrent joint optimization to seamlessly align 2D knowledge with 3D geometry, ensuring coherent consistency across the 3D domain.

3D Scene Understanding. Current 3D scene understanding methods broadly categorized into two groups: (1) methods that operate on known 3D geometry—such as point clouds, depth maps, or meshes; and (2) methods that infer scene semantics while reconstructing the 3D geometry. Methods like [30, 40] and [55, 3] extract semantics via 2D-to-3D lifting, but all depend on pre-built 3D geometry and costly semantic annotations. Recent approaches address this limitation by jointly reconstructing and segmenting 3D scenes through differentiable rendering. NeRF [16, 2] and 3DGS-based methods [52, 65, 31, 18] leverage pseudo-labels to jointly optimize appearance and semantics via 2D supervision. However, both types of methods are inherently offline, relying on full scene observations before reconstruction and interpretation. In real-world settings, agents dynamically explore and progressively understand scenes. To address this gap, we propose an online framework for simultaneous scene reconstruction and understanding. Our method efficiently builds 3D objects while delivering high-quality semantic interpretation. Guided by evolving 3D geometry, it enables comprehensive extraction of open-world objects.

Online Reconstruction. Recent advances in 3DGS [15, 54] have demonstrated remarkable capabilities in photo-realistic rendering and have been extended to a range of downstream applications, including robotic manipulation [59, 24, 37], dynamic scene reconstruction [43, 63, 12, 50], and 3D content generation [5, 64, 34]. However, vanilla 3DGS requires prolonged optimization and offline training with access to full video sequences, limiting its practicality in real-world scenarios.

To address these limitations, recent methods [39, 10, 48, 20] have proposed streaming extensions of 3DGS that significantly reduce training time and memory consumption. However, they rely on multi-view videos and pre-computed global poses, which are often impractical in real-world settings. SLAM-based approaches [26, 19] also enable online scene reconstruction but rely on sparse keyframe tracking and expensive post-refinement, limiting their ability to capture fine-grained geometry and semantics. In a related effort, an online Gaussian-based method [45] has been proposed for scene occupancy prediction. However, it is tailored for a specific task, fails to achieve photo-realistic rendering, and suffers from prohibitively expensive training costs. To overcome these challenges, we propose a novel online Gaussian optimization strategy based on knowledge feature guidance, enabling joint reconstruction and understanding of scenes in an on-the-fly manner.

3 Method

As shown in Fig. 2, the proposed ExtractAnything3D (EA3D) enables open-world 3D object extraction through three key components: **(a)** Knowledge extraction and integration, leveraging VLMs and multi-level VFMs for open-world understanding, integrating knowledge feature maps with an online cache and dynamically embed them into Gaussians via a feedforward way (Sec 3.1). **(b)** Online visual odometry for fast pose estimation and geometric initialization, along with online feature Gaussians that incrementally reconstruct object geometry and transfer knowledge online (Sec 3.2). **(c)** Joint optimization that continuously updates 3D object representations by fusing current observations with historical features (Sec 3.3). EA3D supports a wide range of 3D tasks.

https://doi.org/10.52202/085713-1833

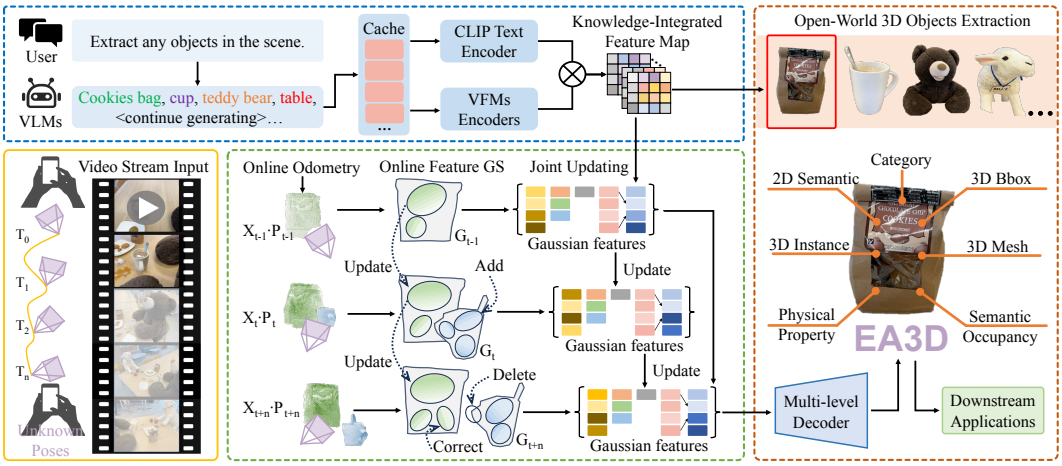


Figure 2: **Framework of EA3D.** Given a streaming video without poses or labels, EA3D first leverages VLMs to identify all potential objects and their physical attributes, while maintaining a dynamic semantic cache to track newly emerging categories. We then use multi-level VFMs to extract knowledge-integrated feature maps from each frame and embed them into Gaussian primitives via a feedforward way. We perform online visual odometry estimation, and incrementally reconstruct geometry and infer knowledge through our online feature Gaussians. A recurrent joint optimization fuses current observations with historical features to continuously update the Gaussians. EA3D supports a wide range of 3D perception tasks and shows strong potential for downstream applications.

3.1 Knowledge-Integrated Feature Map

Given a streaming video, we first extract object-level knowledge by dynamically interpreting the scene frame by frame using 2D vision foundation models (VFMs). However, current 2D foundational vision models lack geometric awareness of 3D scenes, leading to significant multi-view inconsistencies and ambiguities, especially in occluded regions. To tackle this challenge, we propose implicitly aligning foundational visual features in 3D space through a multi-view reconstruction pipeline based on Gaussian Splatting (GS). Each 3D representation primitive is embedded within a knowledge-integrated feature map, utilizing a feed-forward online update strategy.

Open-world interpretation by VLMs. VLMs [13, 57, 42] have shown exceptional open-world understanding in 2D images. Given an image I observed at timestep t, we first use VLMs to identify all instances and their semantics within the image. In an open-world scene, the number and categories of objects are unknown. We use the prompt "Find and list all the possible objects in the given image" to capture any potential objects. Considering the continuously evolving number and semantics of objects in a streaming video, we dynamically maintain an online semantic cache Ω. The online semantic cache takes input of class prompts from VLMs of the current frame, updates the semantics of newly emerged objects, and embeds them into a continuous vector $T \in \mathbb{R}^{1\times V}$ using a pretrained text encoder from CLIP [60, 53], where V denotes the changeable dimension of the vector space.

Semantic feature map. Despite VLMs providing comprehensive open-world interpretation, they exhibit poor visual localization ability. To address this, we leverage foundational vision models [9, 29, 33] to obtain pixel-level segmentation masks and visual features. Given a newly observed image and the online semantic cache, we utilize a pretrained CLIP visual encoder [9] and the Grounded-SAM encoder [35] to generate pixel-wise latent visual feature representations corresponding to each semantic. However, these features contain non-negligible noise and redundant information, which interfere with instance-level segmentation. Therefore, we compute the similarity of each category with semantic features using the embedded continuous vector, generating a binary mask for each category. This mask is then used to aggregate the extracted features using k-nearest neighbors. We then normalize and integrate the semantic features $\mathbf{S} = T \times \mathbf{f}_{sem}$ from different encoders, $\mathbf{f}_{sem}$ denotes the embedded semantic features, and update them into the online semantic cache.

Physical Property. Based on the online semantic cache and 2D priors from VLMs, we also enable the analysis of objects' physical properties. Inspired by [38, 7], we extend the text prompts to extract object-level and part-level physical properties from VLMs, corresponding to the previously obtained

semantics. We then encode the physical attribute features as a variable-length vector $\mathbf{Y}$ with a learnable prompt $y_1, \ldots, y_n$, and fuse it into the online semantic cache.

Feature map embedding. Vanilla Gaussian Splatting [15] represents the geometry through a collection of GS parameters, including position μ, covariance matrix Σ, opacity o, and spherical harmonics coefficients to represent appearance. To synchronize the constructing and understanding of the 3D objects, we add an additional knowledge-integrated feature to each Gaussian. Our method integrates VLM priors, foundational visual features, and inter-track cues, combining the strengths of both appearance and geometry. Specifically, we employ a fast feedforward step to embed the knowledge features encoded by visual foundational models into the Gaussian representations. Retrieved from the online semantic cache and dynamically updated, these knowledge features exchange information across streaming frames over time. Given an emerging video frame I_t at time t, the integrated knowledge feature map $\mathbf{F}_t^{map}$ can be formulated as:

$$\mathbf{F}_t = \sum_{i \in N, j \in N} \mathbf{X}_{\mathbf{i,j}}^{\mathbf{self}} \cdot \mathbf{S}_{\mathbf{i,j}}(\mathbf{T}_{\mathbf{k}}\,; \mathbf{Y}_{\mathbf{i,j}}) \cdot \mathbf{C}_{\mathbf{t}}, \tag{1}$$

where $\mathbf{F}_t$ is the integrated feature map of current frame I_t, $\mathbf{S}_{\mathbf{i,j}}$ denotes the semantic features and $\mathbf{T}_{\mathbf{k}}\,; \mathbf{Y}_{\mathbf{n}}$ are semantic category and physical property tags. i, j denote the pixel coordinates, $\mathbf{X}_{\mathbf{i,j}}^{\mathbf{self}}$ and $\mathbf{C}_{\mathbf{t}}$ represent the corresponding point map and confidence map, as introduced in 3.2. Inspired by [46], we then compute the matching distributions of two consecutive video frames:

$$\mathbf{M}_{t,t-1} = \text{Softmax}\left(\frac{\mathbf{F_t}\mathbf{F_{t-1}}^{\mathsf{T}}}{\|\mathbf{F_t}\|\|\mathbf{F_{t-1}}^{\mathsf{T}}\|}\right), \tag{2}$$

where $\mathbf{F_t}, \mathbf{F_{t-1}} \in \mathbb{R}^{H \times W \times D}$ are the feature maps of two adjacent keyframes, where H, W and D denote height, width and feature dimension, respectively. $\mathbf{M}_{t,t-1} \in \mathbb{R}^{H \times W \times H \times W}$ is the matching distribution between two adjacent keyframes. Based on the guidance from the matching distributions, we continuously propagate the Gaussian features from the previous view to the current frame via a single forward warping, along with their corresponding knowledge feature maps. This ensures the continuity of knowledge transfer through a simple yet effective forward Gaussian transformation. We further provide a detailed comparison of our knowledge-integrated feature embedding against existing feature Gaussian methods [32, 61, 62] in the Appendix.

Multi-level decoder for downstream tasks. Benefiting from the knowledge-integrated feature map, the Gaussian features achieve a unified representation of object geometry and semantics. We then employ a multi-level decoder to decode the Gaussian primitives into diverse outputs, including appearance (i.e., RGB), semantics, physical properties, 3D position, depth map, 3D bounding boxes, and semantic occupancy.

3.2 Online 3D Objects Extraction

Suppose we are walking into a room—the construction and understanding of the 3D space begin the moment we step inside and continuously evolve as we explore. To enable this capability, we propose online feature Gaussians, which support incremental extraction of both geometry and knowledge of 3D objects in an online manner. This framework comprises two core components: 1) **Online visual odometry**, which iteratively generates and updates the poses as new frames are observed; 2) **Online Gaussian updating**, which leverages past observations to rapidly reconstruct and understand the current scene, while dynamically correcting previous misconceptions based on new observation.

Online Visual Odometry. Given an RGB video stream $\{I_t\}_{t=0}^{N}$ without camera pose, we first incrementally estimate the camera pose of the current frame based on a regression of the keypoint graph $(\mathcal{V}, \mathcal{E})$. Each graph node $\mathcal{V}_t$ corresponds to the frame I_t at timestep t, and contains the 6-DoF pose P_t, pointmap X_t, and inverse depth D_t. The graph edges $\mathcal{E}$ denotes the correlation between the current frame and historical frames, with corresponding confidence maps C_t. We use Cut3R [41], a learning-based odometry method, in combination with [23] to estimate the initial pointmap and confidence map. Unlike concurrent work [26, 19], we integrate the dense pixel-level point map generated by Cut3R with sparse points from [23] to more effectively capture the tiny objects in the scene. However, the poses estimated by Cut3R introduce noticeable biases and errors, which accumulate over time. Therefore, we maintain an online keypoint graph and iteratively update it during reconstruction as new frames are processed. Inspired by the local bundle adjustment

https://doi.org/10.52202/085713-1833

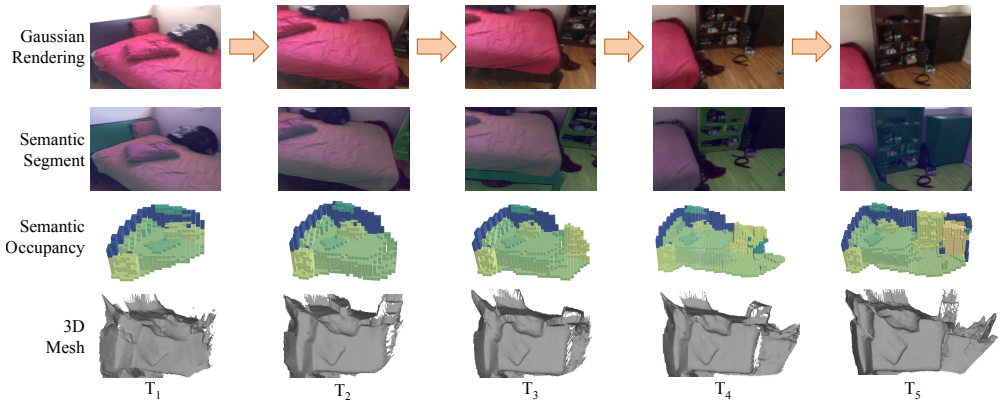


Figure 3: **Visualization of online Gaussian on Scannet [6].** EA3D processes streaming video to incrementally reconstruct while understanding. Historical features guide fast reasoning of current semantics and geometry, while new observations recurrently refine ambiguities and occlusions.

optimization [28] problem, we use a cost function adopted from [4] over the keypoint graph to minimize the reprojection error and update poses for the current frame.

Online Gaussian Updating. Streaming video enables dynamic observation of 3D objects through continuously emerging views, allowing previously under-observed regions to be completed and occlusion-induced ambiguities to be resolved. Inspired by this, we incrementally add feature Gaussians per frame to refine existing geometry and extract new objects. Our approach builds upon HiCoM [10], a streaming GS method designed for multi-view video reconstruction, but overcomes its reliance on predefined poses and multi-view inputs, making it suitable for fully online settings while addressing geometric and semantic challenges.

To overcome these limitations, we develop a semantics-aware online Gaussian update strategy that incrementally adds and adjusts Gaussians based on historical memory and current observations. We initialize Gaussians at timesteps 0 and 1. For each new frame, we back-project the online-estimated inverse depth map D_t and pointmap X_t into 3D to obtain an initial point cloud Φ for object $O \in \Omega$, which is used to initialize the corresponding Gaussians. To reduce redundancy, we adopt the transition strategy from [10, 39], assigning each Gaussian a shared translation vector and rotation quaternion within co-visible regions to maintain inter-frame consistency. For newly observed areas, we introduce new Gaussians with means μ_i initialized from the point cloud, while other attributes are optimized directly. Due to changes in occlusion, some high-opacity ellipsoids may emerge that no longer contribute to specific 3D objects, and we remove them accordingly. Additionally, we apply a one-step splitting strategy to enable adaptive Gaussian growth based on gradients, improving the representation of under-reconstructed regions. Gradients from the entire scene are finally backpropagated to jointly optimize both Gaussian parameters, features and camera poses.

3.3 Recurrent Joint Optimization

During online 3D object extraction, geometric reconstruction and scene understanding mutually reinforce each other. Scene knowledge priors guide the model to focus on areas of interest, while detailed geometry aids in correcting spatial inconsistencies in the priors. Notably, our method enables online joint optimization, without the need for additional post-refinement [26, 19].

Semantic-aware adaptive Gaussian. To leverage the correlation between object semantics and geometry, we design an adaptive semantic-awareness regularization to guide Gaussian scale adjustment:

$$\mathcal{L}_\delta = \sum |\delta_i - \bar{\delta}| F^q_{sem}, \tag{3}$$

where δ_i is the scale of the i-th Gaussian, and $\bar{\delta}$ is the mean scale of the particular semantic Gaussians, F^q_{sem} denotes the semantic feature map corresponding to the q-th object in the semantic cache Ω. The semantic-awareness regularization term encourages Gaussians of the same category to share similar scales, thereby reducing computational overhead caused by redundant scales. After optimizing the

Table 1: Comparison results on ScanNet [6]. The best results are highlighted in **bold**, and the second-best results are underscored. "∗" indicates the use of the colmap-estimated poses following [52, 31, 32]. "–" indicates that the method does not support the specified task. "Rec., Seg., Bbbox., Occ." denotes four multi-task evaluations: reconstruction quality, instance segmentation, 3D bounding box estimation, and semantic occupancy estimation.

Tasks:				Rec.		Seg.		Bbox.		Occ.	
Method	Input	Online	Pose-free	PSNR	SSIM	mIoU	mAcc	AP	mAP	IoU	mIoU
LangSplat [31]	RGB	✗	✗	18.4	0.69	27.5	51.3	-	-	-	-
GaussianGrouping [52]	RGB	✗	✗	19.6	0.74	32.6	56.9	43.6	24.5	47.4	22.1
FeatureGS [32]	RGB	✗	✗	23.9	0.84	41.1	66.0	51.4	32.7	50.9	31.2
OpenGaussian [44]	RGB	✗	✗	22.1	0.80	35.4	61.7	47.5	28.2	49.1	25.3
InstanceGaussian [18]	Points	✗	✗	24.5	0.83	40.5	65.7	52.3	33.4	53.5	32.8
OpenScene [30]	Points	✓	✗	-	-	42.8	68.6	55.7	34.8	51.8	30.5
EmbodiedSAM [47]	RGB-D	✓	✗	-	-	44.2	71.4	58.1	39.5	55.2	33.0
SAM3D [49]	Points	✓	✗	-	-	39.2	62.3	53.7	29.1	53.3	26.7
Enhanced Baselines:											
HiCOM [10]+VFM [35]	RGB	✓	✗	22.6	0.82	34.8	61.9	52.5	23.8	42.4	27.9
MonoGS [26]+VFM [35]	RGB	✓	✗	24.3	0.85	36.3	60.5	51.7	27.7	44.5	27.2
EmbodiedOcc [45]+$\mathcal{L}_{RGB}$	RGB	✓	✗	17.6	0.65	29.2	54.8	56.2	35.6	54.6	33.1
FeatureGS [32]+HiCOM [26]	RGB	✓	✗	24.5	0.85	40.8	66.3	55.8	34.7	50.7	31.4
EA3D*	RGB	✓	✗	25.5	0.87	45.9	71.2	**59.2**	39.6	55.0	**34.3**
EA3D	RGB	✓	✓	**25.8**	**0.89**	**46.3**	**71.8**	57.9	**39.9**	**55.4**	33.9

integrated Gaussian features, we perform alpha-blending to accumulate the final splatted feature $\hat{F}$:

$$\hat{F} = \sum_{i \in N} F_i \cdot \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{4}$$

where α_i denotes the opacity, F_i is the integrated feature map of the i-th Gaussian.

Joint Semantic-geometry Optimization. During online Gaussian training, we jointly optimize Gaussian features and camera poses using a combination of photometric loss, geometric loss, knowledge-integrated loss, and regularization terms, formulated as:

$$\mathcal{L} = \sum_{t=0}^{t_{now}} \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_d + \lambda_3 \mathcal{L}_{kw} + \mathcal{L}_\delta, \tag{5}$$

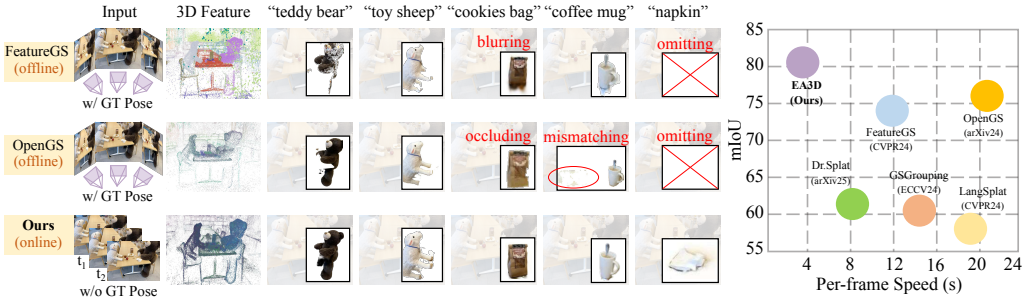
where $\mathcal{L}_1$ is the L_1 photometric loss. $\mathcal{L}_d = \sum |\hat{D}_t - D_t|$, where $\hat{D}_t$ denotes the rendered depth from Gaussian splatting. $\mathcal{L}_{kw}$ denots the L_2 distance between knowledge-integrated feature map and rendered feature map. λ_1, λ_2, and λ_3 are the weighting factors to balance the loss terms. t_{now} denotes the current time step and t_0 is the initial frame. The loss is dynamically computed on the current frame to update existing Gaussian parameters and features, while future frames remain unseen.

4 Experiments

Datasets. We evaluate our method on two benchmarks: LERF [16] dataset comprises in-the-wild scenarios captured with the iPhone App Polycam. The objects in LERF include both common and long-tail categories with different sizes. Scannet [6] is an indoor dataset comprising each annotated with instance-level segmentation and labels across 200 categories. We use 10 RGB sequences selected by [30] without using the depth ground truth or any human annotations.

Implementation Details. We implement EA3D based on HiCoM with a fixed $\lambda_1 = 0.25$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.15$. Each incoming frame is optimized with 100 motion steps, plus another 100 steps after adding new Gaussians. Every fifth frame is used as a test view. All training and testing data remain unseen to the off-the-shelf pretrained models to ensure a fair evaluation. All experiments are conducted on a single A100 80GB GPU. For more details, please refer to the Appendix.

https://doi.org/10.52202/085713-1833



(a) Comparison of 3D objects extraction quality (b) Comparison of quality and training time

Figure 4: **Visualization performance and model efficiency comparison with state-of-the-art methods.** Left (a): Under the more challenging streaming setting without pose input, EA3D delivers high-quality 3D object reconstruction and rendering. Notably, our method avoids redundant Gaussian features through efficient online updates, enabling more precise and lightweight optimization. Right (b): EA3D strikes a balance between speed and quality, significantly reducing training time while maintaining high-performance scene understanding.

Table 2: Comparisons under sparse views and online incremental settings on LeRF [16]. The best results are highlighted in **bold**, and the second-best results are underscored. "–" indicates methods do not support the specified task. "colmap" denotes offline pose estimation using COLMAP, "self." refers to online self-estimated poses. "Speed" denotes the average per-frame optimization speed.

	Tasks:			Rec.(PSNR ↑)			Seg.(mIoU ↑)		
Method	Online	Pose	Speed.(FPS)	10 views	30 views	70 views	10 views	30 views	70 views
LangSplat [31]	✗	colmap	0.007	11.3	14.4	17.8	28.6	34.4	51.5
FeatureGS [32]	✗	colmap	0.018	15.2	18.9	22.4	29.4	41.2	53.6
OpenGaussian [44]	✗	colmap	0.005	14.9	19.5	22.7	30.1	40.5	55.8
Enhanced Baselines:									
Cut3R [41]+VFM [35]	✓	self.	**0.648**	-	-	-	33.7	26.5	21.9
HiCOM [10]+VFM [35]	✓	colmap	0.102	18.1	18.6	21.5	36.1	39.3	43.3
EA3D	✓	self.	0.235	**21.9**	**21.8**	**23.2**	**53.8**	**55.0**	**57.4**

4.1 Quantitative and Qualitative Comparisons

Our method enables holistic 3D object extraction across diverse tasks, including photo-realistic rendering, instance segmentation, and geometric reasoning (e.g., 3D bounding boxes, semantic occupancy, 3D mesh). We validate the effectiveness of our method through comparisons with state-of-the-art approaches and enhanced baselines in 3D reconstruction and online perception.

Compared with reconstruction-based understanding methods. We compare EA3D with NeRF-based [16] and Gaussian-based [31, 52, 44, 32, 18] approaches for 3D scene reconstruction with understanding. These methods rely on offline training with access to all scene views as input. Notably, the compared baselines also require camera poses from GT or Colmap estimated. For fair comparison, we incrementally replace our estimated poses with those from Colmap (denoted as EA3D*).

Results across multiple specific tasks are presented in Table 1. [52, 18, 31] utilize 2D semantic decoded by SAM as supervisions. While effective in 2D segmentation, this strategy fails to learn continuous 3D semantic-geometric representations. Our primary competitors [32, 44] incorporate semantic features but suffer from excessive redundant Gaussians and fail to achieve efficient joint convergence of geometry and semantics. Moreover, all the aforementioned methods rely on complete prior observations of the 3D space, which severely limits their applicability in real-world scenes. In contrast, EA3D adopts an online training strategy that delivers high-quality reconstruction and understanding, while offering better scalability.

Compared with online 3D scene understanding methods. Two common limitations can be observed across these approaches: 1) reliance on predefined geometry or 3D representations (e.g.,

https://doi.org/10.52202/085713-1833

point clouds, depth maps, meshes); 2) dependence on extensive training with large-scale annotated datasets. As shown in Table 1, our method achieves competitive performance even when compared to models trained specifically for the 3D understanding tasks. [49, 30, 47] utilize SAM to obtain 2D segmentations and project them into 3D space, but suffer from semantic ambiguities and multi-view inconsistency caused by mis-projections. In contrast, our approach jointly optimizes geometry and knowledge without relying on 3D priors, demonstrating the strengths of our unified online framework.

Compared with enhanced baselines. Since our work is the first to enable online joint geometry reconstruction and scene understanding, we enhance existing methods in two ways to serve as stronger baselines: 1) augmenting online reconstruction methods with scene understanding capabilities (e.g., HiCOM+VFM, MonoGS+VFM); 2) enabling online optimization of feature Gaussians (e.g., FeatureGS+HiCOM). Additionally, we incorporate an L_1 RGB loss into EmbodiedOcc [45], which was originally designed for online occupancy prediction. Table 1 demonstrates that EA3D consistently outperforms our baseline HiCOM by integrating VFM-driven scene understanding. It also surpasses FeatureGS+HiCOM, which similarly employs semantic features and online updates, highlighting the effectiveness of our unified framework. Furthermore, compared to online SLAM-based methods [45, 26], EA3D achieves better results in both geometric reconstruction and scene interpretation.

Qualitative Comparisons. We further compare the visual quality of 3D object extraction with the baseline methods in Fig. 4(a). Given a streaming video without pose information, EA3D allows high-quality reconstruction and rendering of arbitrary 3D objects. Visualizations of the 3D features show that our online feature Gaussians efficiently and accurately capture both geometry and semantics. In contrast, leading baselines introduce redundant noise, produce inferior renderings, and fail to extract challenging objects (e.g., a small piece of napkin). EA3D also enables a variety of downstream applications, such as manipulation simulation, motion emulation, controllable 3D editing, and object insertion or removal. Additional results and applications are presented in the Appendix.

Our experimental results and theoretical analyses reveal that naïve integrations of existing models tend to perform poorly and may even degrade overall performance due to inherent conflicts among components. In contrast, our method fully harnesses the open-vocabulary features extracted by VFMs and effectively tackles the key challenges of 3D semantic consistency and online geometric reconstruction. Moreover, it achieves higher efficiency and lower computational overhead through a unified and elegantly designed framework.

4.2 Sparse Views and Online Stability

Table 2 reports the performance and robustness of EA3D under sparse-view and online incremental settings. We evaluate it by sequentially inputting sparse-view images (e.g., 10 views) and progressively extending the sequence length. In contrast, offline baselines [31, 32, 44] receive all training views at once. Results show that our method exhibits strong robustness to sparse-view inputs, achieving promising results even with a few initial frames in the early stage. As the sequence length increases ($10 \rightarrow 30 \rightarrow 70$ views), EA3D maintains stable quality, while baseline methods struggle with instability and slow convergence under sparse inputs. Fig. 3 further illustrates the online updating process of rendering and segmentation, occupancy estimation, and 3D mesh generation with EA3D.

Table 3: Ablation on key components. "Train" and "Render" represent the per-frame training and rendering time, measured in FPS. "regular.term" denotes the semantic-awareness regularization. "online.opt", "online.odo", and "joint.opt" denote the online updating strategy, online visual odometry, and joint optimization, respectively.

Strategy	PSNR	mIoU	mAcc	Train	Render
Baseline: HiCoM [10]	22.6	34.8	61.9	0.29	230
W/o CLIP Encoder	25.3	41.6	66.4	0.28	220
W/o SAM Encoder	25.4	42.8	67.1	0.27	215
W/o regular.term	25.1	44.3	70.5	0.21	208
W/o online.opt	24.6	44.5	69.7	0.07	110
W/o online.odo	25.0	45.4	70.8	0.26	205
W/o joint.opt	24.8	45.7	71.4	0.25	210
Ours-full	25.8	46.3	71.8	0.23	210

https://doi.org/10.52202/085713-1833

4.3 Model Efficiency Analysis

Our method enables online incremental reconstruction and understanding of scenes for 3D object extraction. Here, we quantitatively evaluate the speed and memory usage of each key component. As shown in Fig. 4(b), our method achieves faster optimization while maintaining top performance. EA3D strikes a balance between speed and accuracy, delivering higher rendering efficiency with reduced storage overhead. Detailed quantitative experimental results are provided in the Appendix.

4.4 Ablation Studies

As shown in Table 3, we conduct ablation studies and analyze the key components of our designs for online open-world 3D object extraction. Embedded visual features from VFMs (e.g., CLIP [9] and SAM [33]) imbue Gaussians with semantic awareness, enhancing both fine-grained geometry modeling and scene understanding. Our online optimization strategy accelerates feature Gaussian refinement via an efficient feedforward mechanism, ensuring accuracy while minimizing redundancy. The online visual odometry provides dynamic pose updates and dense geometric cues, speeding up convergence. Semantic-aware regularization links Gaussian geometry with semantic features, ensuring object-level 3D consistency and smoothness. By jointly optimizing geometry, semantics, and pose, our method enables recurrent feature updates that seamlessly integrate appearance and structure for robust 3D reconstruction and understanding. For more ablation studies on key modules and hyperparameters, please refer to the Appendix.

5 Conclusion

We have presented EA3D, a unified online framework for open-world 3D object extraction. EA3D enables simultaneous online reconstruction and understanding without geometric or pose priors. It consistently achieves good performance across a broad set of tasks, including photo-realistic reconstruction and rendering, semantic and instance segmentation, 3D bounding box construction, semantic occupancy estimation, and 3D mesh generation. EA3D introduces a novel perspective for aligning and aggregating 3D semantic and geometric features through online reconstruction and dynamic update strategies. It establishes a unified online 3D feature aggregation framework grounded in reconstruction constraints, enabling more accurate and efficient 3D scene understanding and reconstruction.

Acknowledgment

This work was supported by National Key R&D Program of China (Grant No. 2022ZD0160305). This work was also a research achievement of Key Laboratory of Science, Technology, and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). Ming-Hsuan Yang was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project).

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] Yuto Asano, Naruya Kondo, Tatsuki Fushimi, and Yoichi Ochiai. From geometry to culture: An iterative vlm layout framework for placing objects in complex 3d scene contexts. *arXiv preprint arXiv:2503.23707*, 2025. 2

[2] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *NeurIPS*, 36:25971–25990, 2023. 3

[3] 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. 3

[4] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *CVPR*, pages 24–34, 2023. 6

[5] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 3

[6] 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. 6, 7

[7] Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. Pie-nerf: Physics-based interactive elastodynamics with nerf. In *CVPR*, pages 4450–4461, 2024. 4

[8] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *CVPR*, pages 20796–20805, 2024. 2

[9] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, 132(2):581–595, 2024. 3, 4, 10

[10] 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, 6, 7, 8, 9

[11] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *NeurIPS*, 2023. 2

[12] 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 *CVPR*, pages 4220–4230, 2024. 3

[13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv*, 2024. 2, 3, 4

[14] Li Jiang, Shaoshuai Shi, and Bernt Schiele. Open-vocabulary 3d semantic segmentation with foundation models. In *CVPR*, 2024. 2

[15] 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. 2, 3, 5

[16] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, pages 19729–19739, 2023. 2, 3, 7, 8

[17] 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 *ICCV*, pages 4015–4026, 2023. 3

[18] Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. *arXiv preprint arXiv:2411.19235*, 2024. 3, 7, 8

[19] 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. 3, 5, 6

[20] 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. 3

[21] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv*, 2023. 2, 3

[22] Chuang Lin, Yi Jiang, Lizhen Qu, Zehuan Yuan, and Jianfei Cai. Generative region-language pretraining for open-ended object detection. In *CVPR*, pages 13958–13968, 2024. 3

[23] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *ECCV*, pages 424–440, 2024. 5

 https://doi.org/10.52202/085713-1833

[24] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *ECCV*, pages 349–366, 2024. 3

[25] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 2010. 2

[26] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *CVPR*, pages 18039–18048, 2024. 3, 5, 6, 7, 9

[27] 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

[28] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009. 6

[29] Maxime Oquab, Timothe Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[30] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023. 3, 7, 9

[31] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, pages 20051–20060, 2024. 2, 3, 7, 8, 9

[32] 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. 5, 7, 8, 9

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 4, 10

[34] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 3

[35] 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, 7, 8

[36] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *CVPR*, pages 8931–8940, 2024. 2

[37] Ola Shorinwa, Johnathan Tucker, Aliyah Smith, Aiden Swann, Timothy Chen, Roya Firoozi, Monroe Kennedy III, and Mac Schwager. Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. *arXiv preprint arXiv:2405.04378*, 2024. 3

[38] Yinghao Shuai, Ran Yu, Yuantao Chen, Zijian Jiang, Xiaowei Song, Nan Wang, Jv Zheng, Jianzhu Ma, Meng Yang, Zhicheng Wang, et al. Pugs: Zero-shot physical understanding with gaussian splatting. *arXiv preprint arXiv:2502.12231*, 2025. 4

[39] 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, 6

[40] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023. 3

[41] 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. 5, 8

[42] Weihan 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. 3, 4

[43] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, pages 20310–20320, 2024. 3

[44] 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. 7, 8, 9

[45] Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding. *arXiv preprint arXiv:2412.04380*, 2024. 3, 7, 9

[46] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, pages 8121–8130, 2022. 5

[47] Xiuwei Xu, Huangxing Chen, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Embodiedsam: Online segment any 3d thing in real time. *arXiv preprint arXiv:2408.11811*, 2024. 7, 9

[48] 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

[49] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2, 7, 9

[50] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 3

[51] Lewei Yao, Renjie Pi, Jianhua Han, Xiaodan Liang, Hang Xu, Wei Zhang, Zhenguo Li, and Dan Xu. Detclipv3: Towards versatile generative open-vocabulary object detection. In *CVPR*, pages 27391–27401, 2024. 3

[52] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, pages 162–179, 2024. 2, 3, 7, 8

[53] 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. 4

[54] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, pages 19447–19456, 2024. 2, 3

[55] Hongjia Zhai, Hai Li, Zhenzhe Li, Xiaokun Pan, Yijia He, and Guofeng Zhang. Panogs: Gaussian-based panoptic segmentation for 3d open vocabulary scene understanding. *arXiv preprint arXiv:2503.18107*, 2025. 3

[56] Dingyuan Zhang, Dingkang 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. 2

[57] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv*, 2023. 2, 3, 4

[58] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 2

[59] Yuhang Zheng, Xiangyu Chen, Yupeng Zheng, Songen Gu, Runyi Yang, Bu Jin, Pengfei Li, Chengliang Zhong, Zengmao Wang, Lina Liu, et al. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 2024. 3

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 4

[61] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, 2024. 5

[62] Shijie Zhou, Hui Ren, Yijia Weng, Shuwang Zhang, Zhen Wang, Dejia Xu, Zhiwen Fan, Suya 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. 5

 https://doi.org/10.52202/085713-1833

[63] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, pages 21634–21643, 2024. 3

[64] Xiaoyu Zhou, Xingjian Ran, Yajiao Xiong, Jinlin He, Zhiwei Lin, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Gala3d: Towards text-to-3d complex scene generation via layout-guided generative gaussian splatting. *arXiv preprint arXiv:2402.07207*, 2024. 3

[65] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *IJCV*, 133(2):611–627, 2025. 2, 3

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]

https://doi.org/10.52202/085713-1833

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 4.

   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 4.

   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).

https://doi.org/10.52202/085713-1833

- 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 4.

   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.

https://doi.org/10.52202/085713-1833

- 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 in Section 3.1.

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.

https://doi.org/10.52202/085713-1833