X-GS: An Extensible Framework for Perceiving and Thinking with 3D Gaussian Splatting

Yueen Ma[1,3], **Zenglin Xu**[2,3], **Irwin King**[1]
[1]The Chinese University of Hong Kong
[2]Fudan University
[3]Shanghai Academy of AI for Science
mayueen@link.cuhk.edu.hk, zenglinxu@fudan.edu.cn, king@cse.cuhk.edu.hk

Abstract

3D Gaussian Splatting (3DGS) has emerged as a powerful technique for novel view synthesis, subsequently extending into numerous spatial AI applications. However, most existing 3DGS methods operate in isolation, focusing on specific domains. In this paper, we introduce X-GS, an extensible framework that integrates previously isolated 3DGS methods into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*. The *Perceiver* performs online 3DGS-based SLAM with semantic distillation and outputs semantic Gaussians from unposed video streams. It leverages recent vision foundation models for stronger geometric priors, and we introduce three novel optimizations to improve semantic distillation efficiency. The *Thinker* interfaces diverse VLMs with these semantic Gaussians, unlocking spatial multimodal capabilities such as 3D visual grounding and scene captioning. Experimental results on diverse benchmarks demonstrate the efficiency and newly unlocked multimodal capabilities of the X-GS framework.

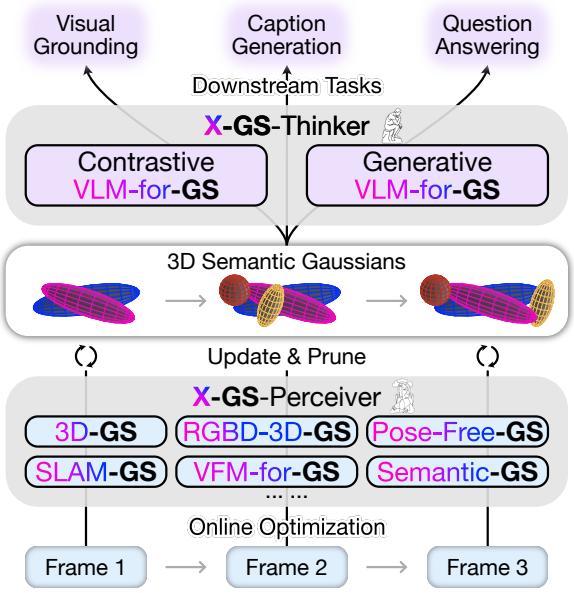


Figure 1: X-GS is an extensible framework that integrates previously isolated 3DGS methodologies into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*. X-GS-*Perceiver* is an online 3DGS-based SLAM system with semantic distillation that outputs semantic Gaussians. X-GS-*Thinker* interfaces diverse VLMs with these semantic Gaussians, unlocking spatial multimodal capabilities.

1 Introduction

The groundbreaking success of 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) in real-time novel view synthesis has sparked rapid advancements across several distinct research directions. To overcome the reliance on computationally expensive offline Structure-from-Motion (SfM) pipelines like COLMAP, recent methods (Fu et al., 2024b) can reconstruct scenes directly from unposed image sequences. Concurrently, the advent of 3DGS-based SLAM systems (Matsuki et al., 2024) has improved upon traditional point-cloud-based methods, allowing the mapped scene to retain significantly richer visual fidelity. Extending beyond pure appearance, semantic-GS (Li et al., 2025c) methods endow the 3D Gaussians with semantic feature channels to facilitate complex scene understanding. Finally, to bridge the gap between spatial representations and multimodal reasoning, the latest works have introduced Vision-Language Models (VLMs) (Thai et al., 2025) that natively process 3D Gaussians as visual inputs.

Despite this rapid progress, these advancements have typically been developed in isolation. To combine their respective strengths, we introduce X-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 21281–21296
October 24-29, 2026 ©2026 Association for Computational Linguistics

GS, an extensible framework designed to integrate these disparate 3DGS techniques into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*, as illustrated in Figure 1. The name stands for an e**X**tensible framework for 3D**GS**, where the "X" serves as a placeholder prefix denoting distinct paradigms such as pose-free-GS, VFM-for-GS, SLAM-GS, semantic-GS, and VLM-for-GS. By synthesizing these diverse capabilities, X-GS achieves online 3DGS-based SLAM with semantic distillation while simultaneously bridging the gap to downstream multimodal VLMs. Importantly, X-GS serves as a foundational blueprint; while this paper presents one concrete instantiation, its extensible nature ensures that as individual research directions evolve, advanced variants of X-GS can be readily integrated.

To realize this integrated vision using currently available 3DGS methods, our concrete instantiation of the framework is structured around these two components. The first, **X-GS-*Perceiver***, is an online 3DGS-based SLAM system with semantic distillation. It jointly models the camera pose, geometry, appearance, and semantics of an environment from unposed video streams and outputs semantic Gaussians. Functioning as an online SLAM system, the *Perceiver* leverages recent Vision Foundation Models (VFMs) (Wang et al., 2025) to extract robust camera poses and dense geometric priors, which are then used to seed the 3D Gaussians. Simultaneously, it distills high-dimensional semantic features directly into these Gaussians. To reduce the computational cost of this semantic distillation during online mapping, we introduce three efficiency optimizations: an EMA-updated vector quantization (VQ) module, GPU-accelerated grid sampling for semantic supervision, and a meticulously parallelized system pipeline.

The second component, **X-GS-*Thinker***, is a versatile module that interfaces diverse VLMs with the semantic Gaussians output by the *Perceiver*, unlocking spatial multimodal capabilities. For contrastive VLMs, the *Thinker* directly queries the distilled features to facilitate open-vocabulary tasks like 3D visual grounding. For generative VLMs, it leverages the global 3D context to guide the model toward the most informative visual tokens across long video sequences. This effectively reduces temporal redundancy, supporting complex multimodal reasoning tasks such as 3D scene captioning and question answering.

Table 1: **Comparison of X-GS with representative 3DGS methods.**

Method	RGB-Only	RGB-D	Pose-Free	Online-SLAM	Semantics	Generative VLM
Offline 3DGS						
3DGS (Kerbl et al., 2023)	✓	×	×	×	×	×
CF-3DGS (Fu et al., 2024b)	✓	×	✓	×	×	×
Feature 3DGS (Zhou et al., 2024)	✓	×	×	×	✓	×
LangSplat (Qin et al., 2024)	✓	×	×	×	✓	×
LangSplatV2 (Li et al., 2025c)	✓	×	×	×	✓	×
Online 3DGS						
MonoGS (Matsuki et al., 2024)	✓	✓	✓	✓	×	×
LEGO-SLAM (Lee et al., 2025)	×	✓	✓	✓	✓	×
OpenMonoGS-SLAM (Yoo et al., 2025)	✓	×	✓	✓	✓	×
EA3D (Zhou et al., 2025)	✓	×	✓	✓	✓	×
EmbodiedSplat (Lee et al., 2026)	✓	✓	×	✓	✓	×
VLMs for 3DGS						
UniGS (Li et al., 2025b)	✓	×	×	×	✓	×
SplatTalk (Thai et al., 2025)	✓	×	×	×	✓	✓
X-GS (Ours)	✓	✓	✓	✓	✓	✓

In summary, our main contributions are:

- We present X-GS, an extensible framework that integrates previously isolated 3DGS methodologies into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*.
- We propose X-GS-*Perceiver*, an online 3DGS-based SLAM system with semantic distillation that outputs semantic Gaussians from unposed video streams; its efficiency is improved by three optimizations: an online VQ module, grid-sampled supervision, and a parallelized pipeline.
- We introduce X-GS-*Thinker*, a versatile module that interfaces diverse VLMs with the semantic Gaussians output by the *Perceiver*, unlocking spatial multimodal capabilities.
- Extensive experiments demonstrate that our integrated framework achieves state-of-the-art performance in both online semantic 3D mapping and downstream 3D reasoning tasks.

2 Related Work

2.1 3DGS-based Online SLAM

Standard 3D Gaussian Splatting (3DGS) algorithms (Kerbl et al., 2023) rely on offline multi-view image sequences and COLMAP initialization. While subsequent methods like CF-3DGS (Fu et al., 2024b) eliminate the need for pre-computed

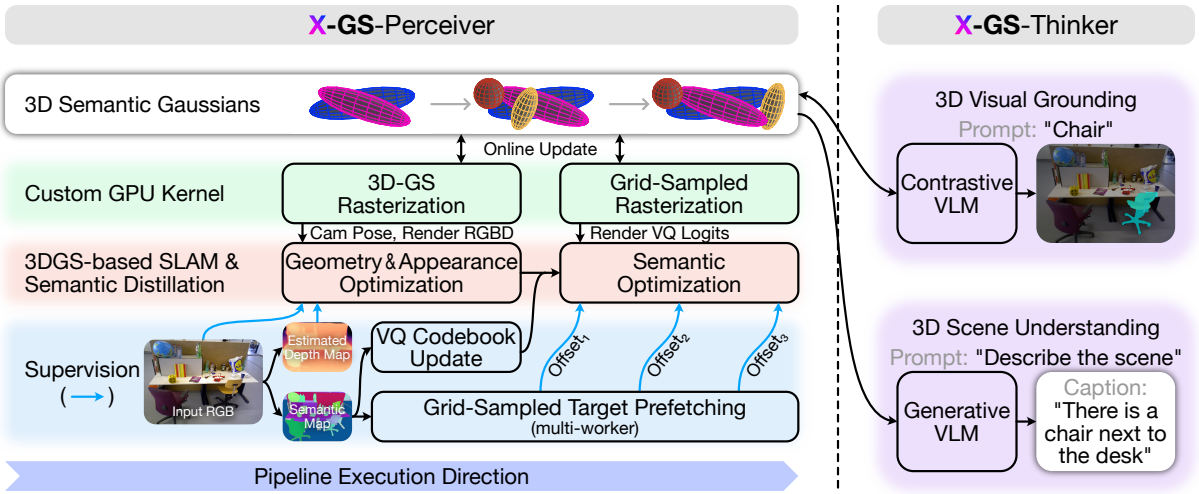


Figure 2: **Overview of the X-GS framework.** The X-GS-*Perceiver* is an online 3DGS-based SLAM system with semantic distillation that outputs semantic Gaussians; a Vector Quantization (VQ) module, grid-based semantic supervision, and an asynchronous parallel pipeline improve its efficiency. Designed as a versatile framework, it can flexibly incorporate various Vision Foundation Models (VFMs). Furthermore, the X-GS-*Thinker* interfaces diverse VLMs with these semantic Gaussians, unlocking spatial multimodal capabilities such as 3D grounding and reasoning.

poses, they still require offline processing. Recently, approaches such as MonoGS (Matsuki et al., 2024), GS-SLAM (Yan et al., 2024), and Gaussian-SLAM (Yugay et al., 2023) have achieved online SLAM using 3DGS. However, these systems focus exclusively on 3D reconstruction and camera tracking, ignoring high-level semantic scene understanding. To enrich 3DGS representations, Feature 3DGS (Zhou et al., 2024) and LangSplat (Qin et al., 2024; Li et al., 2025c) distill dense semantic features from vision foundation models (e.g., SAM or CLIP) directly into the 3D Gaussian field, enabling language-driven queries and zero-shot 3D object detection. Nevertheless, these semantic methods rely heavily on precise, pre-computed camera poses and are strictly designed for offline mapping, inherently limiting their applicability in dynamic or autonomous environments.

2.2 Semantic SLAM

Existing semantic 3DGS SLAM systems, such as LEGO-SLAM (Lee et al., 2025) and OpenMonoGS-SLAM (Yoo et al., 2025), attempt online semantic mapping but suffer from rigid architectural bottlenecks. LEGO-SLAM strictly requires RGB-D input, whereas OpenMonoGS-SLAM operates solely on monocular video—lacking the flexibility to utilize depth maps even when available. Furthermore, OpenMonoGS-SLAM is computationally expensive. X-GS natively overcomes all of these limitations.

2.3 VLMs for 3DGS

Recent literature (Li et al., 2024) demonstrates the immense value of explicitly integrating 3D representations with Vision-Language Models (VLMs) for downstream reasoning. For instance, UniGS (Li et al., 2025b) aligns optimized 3D Gaussians with textual spaces for multimodal contrastive learning, while SplatTalk (Thai et al., 2025) and Chat-Splat (Chen et al., 2024a) build VLMs that take 3DGS as visual input. However, current VLMs for 3DGS are restricted entirely to static, offline scenes. X-GS integrates these disjoint paradigms into a single framework.

Table 1 provides a comparison of key capabilities among representative 3DGS methods. To the best of our knowledge, X-GS is the first framework to fulfill all the listed requirements.

3 Method

We propose X-GS, an extensible framework designed to integrate previously isolated 3DGS advancements into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*. As illustrated in Figure 2, **X-GS-*Perceiver*** is an online 3DGS-based SLAM system with semantic distillation: it ingests unposed monocular video streams (with optional

depth) to continuously co-optimize 3D Gaussians and camera poses while distilling VFM semantic features into them, outputting semantic Gaussians. **X-GS-*Thinker*** is a versatile module that interfaces diverse VLMs with these semantic Gaussians, unlocking spatial multimodal capabilities such as text-prompted 3D visual grounding and zero-shot scene captioning.

3.1 Preliminaries

3DGS represents a scene using 3D Gaussians, each parameterized by $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \mathbf{c}\}$ (center, covariance, opacity, and color). For a camera pose $\mathbf{T} \in \mathrm{SE}(3)$, each 3D Gaussian is projected into a 2D elliptical splat; the expected pixel color $C(x)$ is then computed by front-to-back α-blending N_g depth-sorted splats:

$$C(x \mid \mathbf{T}, \Theta) = \sum_{i=1}^{N_g} \mathbf{c}_i \alpha_i' \prod_{j=1}^{i-1} (1 - \alpha_j'), \quad (1)$$

where α_i' is the evaluated 2D opacity of the i-th splat. This rasterization naturally extends to semantic features: by augmenting the 3D Gaussians with a set of D-dimensional features $\mathbf{f} = \{\mathbf{f}_i\}_{i=1}^{N_g}$, we render a dense 2D semantic feature map $F(x \mid \mathbf{T}, \Theta, \mathbf{f})$ by substituting $\mathbf{c}_i$ with $\mathbf{f}_i$ in the identical blending formulation.

3.2 X-GS-*Perceiver*

X-GS-*Perceiver* is designed to jointly model the geometry, appearance, and semantics of an environment, yielding a comprehensive 3D representation that facilitates various downstream tasks. It is highly extensible, as its semantic features can be distilled from a wide range of Vision Foundation Models (VFMs), such as SAM, CLIP, and SigLIP.

3.2.1 3DGS-based SLAM

3DGS was originally designed to process multi-view images offline. MonoGS (Matsuki et al., 2024) later proposed converting 3DGS into an online SLAM pipeline, operating via two main phases: tracking and mapping. During tracking, camera poses are estimated. During mapping, the 3D Gaussians are optimized to match the input images.

Unlike MonoGS, we leverage recent VFMs that can simultaneously provide accurate spatial information, including camera poses, depth estimates, and point clouds, exemplified by VGGT (Wang et al., 2025). Since the original VGGT is also an offline model, we employ its online variants, such as StreamVGGT (Zhuo et al., 2025) and InfiniteVGGT (Yuan et al., 2026).

Tracking. During tracking, for each incoming frame, we use the online VFM's camera pose estimation for initialization, replacing the zero-velocity assumption used in MonoGS. Although these VFMs provide robust initial camera poses, the poses may still accumulate slight drift over time. Therefore, we further finetune the poses against the input color image I_t and depth map D_t by minimizing the tracking loss $\mathcal{L}_{\text{track}}$:

$$\mathbf{T}_t^* = \arg\min_{\mathbf{T}_t} \mathcal{L}_{\text{track}}(I_t, D_t \mid \mathbf{T}_t, \Theta). \quad (2)$$

Depth supervision is primarily supplied by the VFM, though we can seamlessly accommodate ground-truth input depth maps if available.

Mapping. The mapping phase refines the global scene representation by co-optimizing the Gaussian parameters Θ along with the camera poses $\mathbf{T}$. Specifically, we first initialize new Gaussians from point clouds provided directly by the VFMs or backprojected from their estimated depth maps. This serves as a much stronger geometric prior than naive flat-plane initialization. Furthermore, to mitigate rendering artifacts, we incorporate an occluder pruning mechanism that removes floating Gaussians that are much closer to the camera than the estimated depth, as they can occlude the Gaussians behind and hinder the optimization process.

The mapping optimization is performed over a sliding window of keyframes $\mathcal{K}$. Instead of relying on heuristic 2D overlap ratios, the window is dynamically maintained by inserting new keyframes based on the VFM's robust 4D point tracks and visibility masks, while evicting redundant older views to preserve a maximum capacity of N_{win}. The mapping thread refines Θ by minimizing the mapping loss $\mathcal{L}_{\text{map}}$ across this active window of ground-truth images I_k and depth maps D_k:

$$\Theta^* = \arg\min_{\Theta} \sum_{k \in \mathcal{K}} \mathcal{L}_{\text{map}}(I_k, D_k \mid \Theta, \mathbf{T}_k). \quad (3)$$

To better preserve the VFM's depth map prior, we slow the learning rate for the Gaussians' position updates during this optimization.

3.2.2 Online Semantic Distillation

Geometry and appearance information provided by plain 3DGS cannot satisfy the needs of downstream multimodal tasks. Therefore, we add

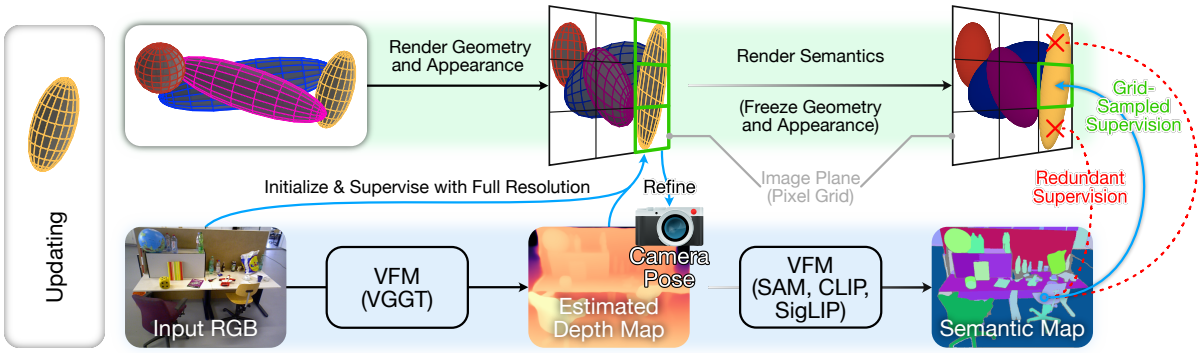


Figure 3: Illustration of the information flow among the core components of X-GS-*Perceiver*, alongside an illustrative example of grid-sampled semantic supervision.

semantic distillation capabilities to the SLAM pipeline. Naively attaching semantic features to 3D Gaussians has proven to be heavily memory- and compute-intensive (Qin et al., 2024). To reduce this cost during online mapping, we introduce three efficiency optimizations: an online Vector Quantization (VQ) module, a GPU-accelerated grid-sampling scheme, and a meticulously designed pipeline that fully leverages highly parallelized scheduling. Figure 3 illustrates the information flow among these components and the grid-sampled semantic supervision scheme.

Online VQ Module. Vector Quantization reduces the dimensionality of semantic features at each pixel. Inspired by LangSplatV2 (Li et al., 2025c), we represent the semantic state of each Gaussian i using a learnable logit vector $\mathbf{z}_i \in \mathbb{R}^L$ over a shared codebook $\mathbf{E} \in \mathbb{R}^{L\times D}$, where the l-th row $\mathbf{E}_{l,*} \in \mathbb{R}^D$ represents a single codeword. These logits are converted into mixture weights via $\mathbf{w}_i = \mathrm{softmax}(\mathbf{z}_i)$. The decoded semantic feature is then obtained by a weighted sum over the codebook: $\hat{\mathbf{f}}_i = \mathbf{E}^\top \mathbf{w}_i \in \mathbb{R}^D$. This parameterization separates Gaussian-specific coefficients from shared semantic codewords.

We update the shared codebook online using an exponential moving average (EMA). Given a set of observed semantic features $\{\mathbf{x}_n\}_{n=1}^N$, each is assigned to its nearest codeword $a_n = \arg\min_l \|\mathbf{x}_n - \mathbf{E}_{l,*}\|_2$. We accumulate the assignment counts $c_l = \sum_n \mathbb{I}[a_n = l]$ and feature sums $\mathbf{s}_l = \sum_n \mathbb{I}[a_n = l]\mathbf{x}_n$. The codebook is then updated via EMA buffers, where N_l tracks the moving cluster size and the matrix $\mathbf{M} \in \mathbb{R}^{L\times D}$ tracks the moving feature sums. For each codeword l, the corresponding EMA statistics and codebook row are updated:

$$\begin{aligned} N_l &\leftarrow \lambda N_l + (1-\lambda)c_l, \\ \mathbf{M}_{l,*} &\leftarrow \lambda \mathbf{M}_{l,*} + (1-\lambda)\mathbf{s}_l, \\ \mathbf{E}_{l,*} &\leftarrow \frac{\mathbf{M}_{l,*}}{N_l + \varepsilon}, \end{aligned} \tag{4}$$

where $\lambda \in [0, 1)$ is the decay rate and ε is a small constant for numerical stability.

Grid-Sampled Semantic Supervision. Grid-sampling aims to reduce the resolution of the semantic map (height and width, i.e., the number of pixels). Applying dense semantic supervision at every image pixel is unnecessarily expensive during mapping, as each 3D Gaussian typically projects to a 2D splat covering many pixels. To reduce this cost, instead of rendering a full-resolution semantic feature map, we supervise the features on a sparse, regular grid defined by a specific stride and offset.

Crucially, rather than relying on standard high-level tensor subsampling, we implement this via **a custom GPU kernel written in CUDA**. This low-level optimization performs only the minimal pixel-level calculations required for the subsampled locations. By doing so, it entirely avoids the computational and memory burdens of handling a full-resolution semantic supervision map.

During optimization, we compute a combined L1 and cosine similarity loss between the rendered sparse features and the ground-truth features. We pair this with a binary validity mask to filter out unannotated background regions (preventing Gaussians from learning empty features) and to handle out-of-bounds coordinates. With stride s, our rasterizer reduces supervised pixels, memory use, and computation by s^2 during online mapping.

Parallelized Pipeline. To integrate these individual components into a highly efficient workflow, we

Table 2: **Quantitative evaluation on SLAM benchmarks.** RGB-D methods (gray numbers) utilize ground-truth depth and are therefore not directly comparable to monocular (RGB-only) baselines or to X-GS-*Perceiver*. Rows highlighted in gray are ablation studies. ATE denotes absolute trajectory error, reported as RMSE (cm). PCD: point clouds. FF: feed-forward. Alt.: alternative depth prior (VGGT-SLAM in place of StreamVGGT).

Method	Paradigm	Tracking	Appearance (RGB)			Geometry (Depth)			
		ATE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑
BENCHMARK: **TUM RGB-D**									
RGB-D									
Photo-SLAM (Huang et al., 2024)	RGB-D SLAM	1.80	19.28	0.673	0.334	—	—	—	—
SEGS-SLAM (Wen et al., 2025)	RGB-D SLAM	1.14	23.20	0.769	0.271	—	—	—	—
Monocular									
SplatTalk (Thai et al., 2025)	Offline FF 3DGS	—	15.51	0.483	0.243	0.795	5.301	7.006	0.000
StreamVGGT (Zhuo et al., 2025)	Online PCD	2.70	13.58	0.281	0.594	0.105	0.076	0.500	0.879
MonoGS (Matsuki et al., 2024)	Online 3DGS	3.96	21.89	0.733	0.327	0.352	0.292	0.666	0.446
X-GS-*Perceiver* (Ours)	Online 3DGS	**1.55**	**25.21**	**0.784**	**0.236**	**0.055**	**0.032**	**0.326**	**0.945**
Ablation Studies									
W/o StreamVGGT	No depth prior	4.49	23.66	0.746	0.256	0.162	0.126	0.616	0.746
With VGGT-SLAM	Alt. depth prior	3.11	23.42	0.754	0.252	0.063	0.047	0.387	0.938
BENCHMARK: **Replica**									
RGB-D									
NICE-SLAM (Zhu et al., 2022)	RGB-D SLAM	1.07	24.42	0.809	0.233	—	—	—	—
Vox-Fusion (Yang et al., 2022)	RGB-D SLAM	3.09	24.41	0.801	0.236	—	—	—	—
Photo-SLAM	RGB-D SLAM	10.63	30.37	0.904	0.161	—	—	—	—
SEGS-SLAM	RGB-D SLAM	9.25	33.54	0.927	0.104	—	—	—	—
Monocular									
SplatTalk	Offline FF 3DGS	—	16.90	0.677	0.431	0.807	11.228	14.416	0.006
StreamVGGT	Online PCD	5.93	7.32	0.044	1.125	0.030	0.013	0.159	0.980
MonoGS	Online 3DGS	2.80	29.31	0.892	0.231	0.155	0.104	0.500	0.763
X-GS-*Perceiver* (Ours)	Online 3DGS	1.12	**34.66**	**0.937**	**0.115**	0.022	0.008	0.094	0.991
Ablation Studies									
W/o StreamVGGT	No depth prior	3.92	29.59	0.889	0.180	0.127	0.077	0.396	0.836
With VGGT-SLAM	Alt. depth prior	**0.81**	33.00	0.926	0.132	**0.016**	**0.003**	**0.076**	**0.996**

employ a pipeline with highly parallelized scheduling, as illustrated in Figure 2. The VQ codebook update is executed as soon as the VFM completes encoding an incoming keyframe. Simultaneously, we initiate a "grid-sampled target prefetching" operation on the semantic map. Because operations across different grid offsets are mutually independent, we allocate multiple background workers to extract and buffer the grid-sampled targets well in advance of the actual optimization loop.

Crucially, to maintain system stability and reduce peak computational load, we decouple the geometry and appearance updates from the semantic updates. During the semantic distillation phase, all other parameters of the 3D Gaussians (i.e., position, scale, rotation, opacity, and color) remain frozen. Conversely, the learned semantic logits and VQ codebooks are strictly excluded from the base tracking and mapping phases. This design reduces distillation costs and supports real-time rendering.

3.3 X-GS-*Thinker*

X-GS-*Thinker* interfaces the resulting semantic Gaussians with diverse VLM architectures, enabling spatial multimodal capabilities.

3.3.1 Contrastive VLM

For 3D visual grounding, the *Thinker* employs contrastive VLMs consisting of an image encoder and a text encoder, such as CLIP (Radford et al., 2021), as well as VFMs for segmentation (Kirillov et al., 2023). Since the 3D Gaussians from the *Perceiver* already encapsulate distilled image embeddings as semantic features, we can directly query and identify target objects by computing the similarity between the Gaussians' semantic features, decoded via the online VQ module, and the prompt embeddings produced by the text encoder.

3.3.2 Generative VLM

To achieve 3D scene understanding, the *Perceiver* distills the embeddings from the vision encoder of a generative VLM, ensuring these semantic features can be directly utilized by the VLM's backbone, acting as the *Thinker*. Because utilizing the full set of N_g Gaussians is computationally prohibitive and yields highly redundant tokens, we condense the scene into a compact sequence of $M \ll N_g$ informative tokens using an Entropy-Adaptive Gaussian Sampling strategy (Thai et al., 2025). Since the underlying 3DGS map is constructed from sequential camera observations, this approach effectively al-

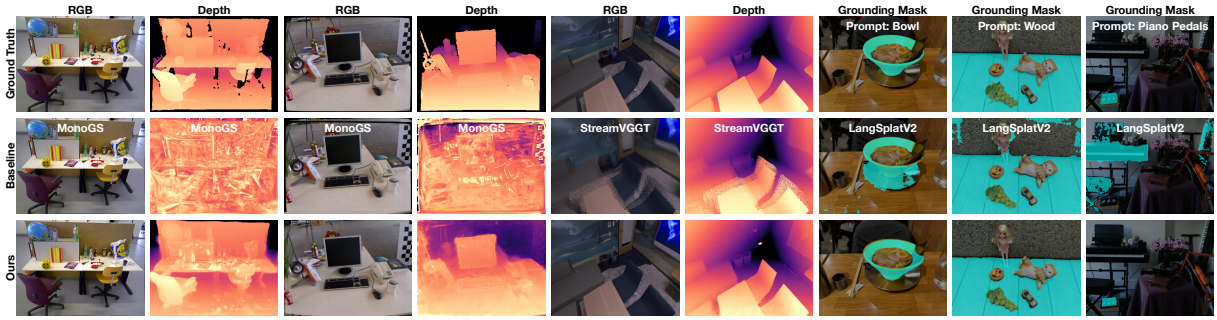


Figure 4: **Qualitative Comparisons.** Visual comparisons of X-GS against baseline methods and ground truth (GT) on scene reconstruction and 3D visual grounding tasks.

Table 3: **Quantitative results on 3D visual grounding benchmarks.** Contrastive X-GS is reported using overall mAcc for LERF Localization and overall mIoU for LERF Segmentation and 3D-OVS. Times in parentheses, taken from Table 5 (CTR), show default → ablation runtimes.

Method	LERF		3D-OVS
	mAcc	mIoU	mIoU
FFD (Kobayashi et al., 2022)	–	–	26.9
LSeg (Li et al., 2022)	21.1	16.6	20.6
LERF (Kerr et al., 2023)	73.6	37.4	54.8
3D-OVS (Liu et al., 2023b)	–	–	86.8
GS-Grouping (Ye et al., 2024)	49.1	36.6	87.7
LEGaussian (Shi et al., 2024)	67.4	24.6	88.5
GOI (Qu et al., 2024)	59.2	42.0	90.6
GAGS (Peng et al., 2024)	81.7	54.1	–
LangSplat (Qin et al., 2024)	84.3	51.4	93.4
LangSplatV2 (Li et al., 2025c)	84.1	59.9	94.6
X-GS	89.7	61.8	94.3
Ablation Studies			
W/o StreamVGGT (With RGB only: 3.04s → 0s)	82.8	56.6	93.5
W/o MobileSAM (With SAM: 3.87s → 73.73s)	**89.9**	61.9	**95.4**
W/o grid-sampling (With full resolution: 4.27s → 37.02s)	89.4	**62.6**	94.8
W/o online distillation	88.9	62.3	94.2
W/o VQ module	(Out of Memory)		

lows X-GS to act as a powerful compression mechanism, distilling long video sequences into a concise set of informative tokens for the VLM.

4 Experiments

4.1 Implementation Details

For the X-GS-*Perceiver*, we utilize a weight of 0.05 for the online VGGT's (Yuan et al., 2026) estimated depth map, and we scale the learning rate of the position parameters by a factor of 0.2. The occluder pruning mechanism uses a relative margin of 10%. The VQ codebook is instantiated with $L = 64$ independent semantic codewords. During the online codebook learning stage, the EMA update incorporates a decay momentum of $\lambda = 0.96$ and a temperature of $\tau = 0.7$ to smoothly adapt the cluster centers to the sequential frame stream. We maintain a sliding window of $N_{win} = 8$ keyframes, allocating 150 optimization iterations for the geometric and photometric mapping phase, followed by 50 iterations dedicated solely to the semantic distillation phase. The *Perceiver* can operate efficiently on a single NVIDIA V100 GPU.

4.2 SLAM Benchmarks

We evaluate X-GS-*Perceiver* on two standard SLAM benchmarks, TUM RGB-D (Sturm et al., 2012) and Replica (Straub et al., 2019), assessing tracking accuracy, photometric rendering quality (RGB), and geometric reconstruction accuracy (depth). Except for the RGB-D SLAM baselines and SplatTalk, which uses posed RGB images, all evaluated models rely solely on monocular RGB inputs and operate without access to ground-truth depth, camera poses, or auxiliary data. Quantitative results are summarized in Table 2.

By leveraging recent VFMs capable of simultaneous camera pose and depth estimation, X-GS-*Perceiver* outperforms MonoGS in both tracking and rendering quality. Furthermore, supervising the 3D Gaussians with the estimated depth maps yields significant improvements in geometric accuracy. Qualitative comparisons are provided in Figure 4.

4.3 3D Visual Grounding Benchmarks

Under the contrastive paradigm of X-GS, we prompt the *Thinker* to identify relevant 3D Gaussians, effectively grounding natural language queries within the perceived 3D scene. We evaluate our method on two 3D visual grounding benchmarks: LERF (Kerr et al., 2023) and 3D-OVS (Liu et al., 2023b). Baselines and quantitative results are reported in Table 3.

Table 4: **Quantitative results on 3D vision-language benchmarks.** The abbreviations C, B-4, M, R, and EM@1 denote CIDEr, BLEU-4, METEOR, ROUGE, and Exact Match, respectively.

Method	Scan2Cap				ScanQA					SQA3D
	C	B-4	M	R	C	B-4	M	R	EM@1	EM@1
Task-Specific Models										
Scan2Cap (GPT-3) (Chen et al., 2021)	35.2	22.4	21.4	43.5	–	–	–	–	–	41.0
3DJCG (Cai et al., 2022)	47.7	31.5	24.3	51.8	–	–	–	–	–	–
Vote2Cap-DETR (Chen et al., 2023)	61.8	34.5	26.2	54.4	–	–	–	–	–	–
ScanRefer+MCAN (Chen et al., 2020)	–	–	–	–	55.4	7.9	11.5	30.0	18.6	–
ClipBERT (Lei et al., 2021)	–	–	–	–	–	–	–	–	–	43.3
ScanQA (Azuma et al., 2022)	–	–	–	–	64.9	10.1	13.1	33.3	21.1	47.2
3D VLMs										
3D-VisTA (Zhu et al., 2023)	66.9	34.0	27.1	54.3	69.6	10.4	13.9	35.7	22.4	48.5
3D-LLM (Hong et al., 2023)	–	8.1	13.1	33.2	69.4	12.0	14.5	35.7	–	–
LL3DA (Task-Specific) (Chen et al., 2024b)	65.2	36.8	26.0	55.1	76.8	13.5	15.9	37.3	–	–
LL3DA (Generalist)	63.0	36.0	25.7	54.7	75.7	13.3	15.4	37.0	–	–
Scene-LLM (Task-Specific) (Fu et al., 2024a)	37.9	24.1	21.8	45.6	80.0	12.0	16.6	40.0	–	54.2
Scene-LLM (Generalist)	–	–	–	–	80.0	11.7	15.8	35.9	–	53.6
2D VLMs										
MoE-LLaVA (Lin et al., 2024)	27.4	18.1	12.1	40.5	42.5	5.0	7.7	21.8	11.3	22.3
LLaVA-NeXT (Liu et al., 2023a)	36.0	20.4	14.2	49.5	48.6	6.3	9.8	22.3	14.6	23.5
LLaVA-OneVision (Zero-Shot) (Li et al., 2025a)	28.3	4.7	11.6	32.4	50.0	–	13.0	29.4	15.6	25.4
2D VLM-Based Models										
SplatTalk (Thai et al., 2025)	–	–	–	–	61.7	–	14.2	32.7	17.1	26.1
SplatTalk-ScanQA-FT	–	–	–	–	77.1	–	15.4	38.7	22.3	42.0
SplatTalk-3DVQA-FT	–	–	–	–	77.5	–	15.6	38.5	22.4	47.6
X-GS (Zero-Shot)	31.8	5.9	12.3	38.5	64.0	7.1	15.1	33.4	19.3	27.0
X-GS (Finetuned)	**68.3**	**38.3**	**27.5**	**55.5**	**83.8**	**15.3**	**17.2**	**41.5**	**23.1**	**54.9**
Ablation Studies (Zero-Shot)										
W/o X-GS-*Perceiver*	27.5	4.5	11.3	31.9	49.2	5.0	12.9	29.2	15.1	24.6
W/o entropy-adaptive sampling (use all tokens)	19.1	0.0	–	22.7	45.4	0.0	–	27.2	9.9	34.0
With spatially uniform sampling	28.2	5.0	11.4	34.6	60.4	6.6	14.0	31.5	18.1	25.3

Our results demonstrate that X-GS outperforms LangSplatV2, an improvement we attribute to the superior geometric representations enabled by the online VGGT model. Qualitative comparisons illustrating this are provided in Figure 4. Furthermore, because LangSplatV2 relies on SAM 1.0 (Kirillov et al., 2023) and requires full-resolution semantic maps for supervision, its training process is computationally expensive (see Table 5). By contrast, our accelerated online semantic distillation strategy, coupled with the integration of MobileSAM (Zhang et al., 2023), dramatically increases processing speed without compromising visual grounding quality.

4.4 3D Vision-Language Benchmarks

To evaluate the 3D understanding and reasoning capabilities of X-GS, we employ three representative 3D vision-language (3D-VL) benchmarks: Scan2Cap (Chen et al., 2021), ScanQA (Azuma et al., 2022), and SQA3D (Ma et al., 2023). Quantitative results are presented in Table 4.

We instantiate the X-GS-*Thinker* using LLaVA-OneVision (Li et al., 2025a), backed by the Qwen2-7B-Instruct LLM (Yang et al., 2024). In this configuration, the *Thinker* interfaces the VLM with semantic Gaussians: it directly ingests 3D Gaussians containing semantic features distilled from the SigLIP (Zhai et al., 2023) VFM, which is the default vision encoder for LLaVA-OneVision. By acting as an efficient compression mechanism for these semantic features, X-GS achieves improved zero-shot performance compared to the base LLaVA-OneVision model. Furthermore, following LoRA (Hu et al., 2022) finetuning, X-GS outperforms specialized 3D VLMs that are natively designed for these 3D-VL tasks.

4.5 Ablation Studies

We ablate the key components of X-GS in the corresponding benchmark tables (gray rows). On SLAM (Table 2), removing StreamVGGT (Zhuo et al., 2025) isolates the geometry prior. For 3D visual grounding (Table 3), we ablate VGGT, MobileSAM, grid-sampling, the VQ module, and online semantic distillation. Table 4 ablates the zero-shot *Thinker*. Replacing reconstructed Gaussians with 2D tokens lowers accuracy across benchmarks, while entropy-adaptive sampling from the 3D map remains comparatively strong and outperforms spatially uniform sampling, indicating that the spatial structure of 3DGS is beneficial for 3D vision-language tasks. Runtime is analyzed in Section 4.6.

These results also provide experimental evidence of extensibility: alternative modules can be integrated into the same X-GS slots. VGGT-

Table 5: Computational footprint analysis of X-GS-*Perceiver*, run on a single NVIDIA V100 GPU. For X-GS, we report the two variants separately: contrastive VLM (CTR), evaluated on TUM RGB-D, and generative VLM (GEN), evaluated on ScanNet. Paired times are for single-thread (ST) and multi-thread (MT) settings.

Main Component	**Baseline**		**X-GS** (CTR)		**X-GS** (GEN)	
	Module	**Time** ST | MT	**Module**	**Time** ST | MT	**Module**	**Time** ST | MT
Geometry & Appearance Opt.	(Baseline: MonoGS)					
VFM Depth Est.	–	–	Online VGGT	3.04	Online VGGT	0.78
Forward Pass	RGB	10.40 | 0.76	RGB-D	11.52 | 0.85	RGB-D	8.76 | 0.65
Backward Pass	RGB	2.77 | 0.20	RGB-D	3.09 | 0.23	RGB-D	4.25 | 0.30
Semantic Supervision Prep.	(Baseline: LangSplatV2)					
Vision Encoding	SAM + CLIP	73.73	MobileSAM+CLIP	3.87	SigLIP+Proj.	0.11
VQ Codebook	Offline	0.06	Online EMA	0.04	Online EMA	0.91
Target Construction	Offline	25.87	Multi-Worker Prefetch	0.40	Multi-Worker Prefetch	1.34
Semantic Optimization	(Baseline: LangSplatV2)					
Forward Pass	Full-Resolution	25.41	Grid-Sampled	3.37 | 0.71	Grid-Sampled	4.19 | 0.93
Backward Pass	Full-Resolution	11.61	Grid-Sampled	16.80 | 3.56	Grid-Sampled	21.80 | 4.59
Overall	(Baseline: MonoGS)					
Keyframe Interval		4.63		4.17		14.20
Asynchronous Overlap		0.00 | 0.00		4.31 | 1.08		2.36 | 0.95
Time per Keyframe		13.17 | 2.41		37.82 | 10.22		39.78 | 11.93
Time per Frame		2.84 | 0.52		9.07 | 2.45		2.80 | 0.84
Execution FPS		0.35 | 1.93		0.11 | 0.41		0.36 | 1.19
Rendering	(Baseline: LangSplatV2)					
RGB FPS		304.90		312.50		–
Semantic FPS		85.60		105.40		–

SLAM can be used in place of StreamVGGT at the *Perceiver* pose/depth interface (Table 2). Unlike StreamVGGT, VGGT-SLAM is submap-based rather than per-frame streaming. SAM 1.0 can be used instead of MobileSAM for semantic segmentation (Table 3). CLIP and SigLIP are both compatible with X-GS's semantic distillation, supporting the contrastive *Thinker* in Table 3 and the generative *Thinker* in Table 4, respectively.

4.6 Computational Footprint Analysis

Table 5 details the runtimes of the contrastive (CTR) and generative (GEN) variants of X-GS-*Perceiver*. We report both a single-thread and a multi-thread setting. The reconstructed 3D semantic Gaussians can be rendered in real time: contrastive X-GS attains 312.5 RGB FPS and 105.4 semantic FPS. Our optimizations focus primarily on accelerating online semantic distillation. Substituting SAM 1.0 (Kirillov et al., 2023) with MobileSAMv2 (Zhang et al., 2023) reduces segmentation time from over 70 seconds to approximately 4 seconds, while our grid-sampled semantic supervision strategy accelerates the forward and backward passes several-fold.

Furthermore, by executing vision encoding and supervision map construction concurrently with geometry and appearance optimization, we effectively hide these computationally expensive steps. Consequently, X-GS reduces the semantic distillation overhead from several minutes per keyframe, as seen in LangSplatV2, to just a few seconds.

5 Conclusion

We propose X-GS, an extensible framework that integrates previously isolated 3DGS methods into the perception module of a VLM for spatial tasks, with two major components: the *Perceiver* and the *Thinker*. While it serves as a general blueprint, we also present a concrete instantiation using current techniques. Our *X-GS-Perceiver* is an online 3DGS-based SLAM system with semantic distillation that outputs semantic Gaussians; its efficiency is improved by an EMA-updated VQ module, GPU-accelerated grid sampling, and system-wide parallelization. By combining VFM-based pose and depth priors with differentiable 3DGS optimization, the *Perceiver* reconstructs explicit, queryable scene representations directly from unposed video. These efficiency mechanisms allow geometry, appearance, and semantics to be updated within a unified online pipeline. Furthermore, the *X-GS-Thinker* interfaces diverse VLMs with the semantic Gaussians output by the *Perceiver*, unlocking spatial multimodal capabilities. Its contrastive interface supports language-grounded 3D localization, while its generative interface condenses semantic Gaussians into informative tokens for scene captioning and question answering. Together, these instantiations show that the X-GS framework can connect online semantic 3DGS maps to both contrastive and generative VLMs, while allowing each pathway to retain the semantic encoder and representation required by its downstream task.

Limitations

First, X-GS currently consists of two main modules: the *Perceiver* and the *Thinker*. Future work could explore end-to-end architectures. Second, as demonstrated by recent systems such as FreeSplat (Wang et al., 2024), SplatTalk (Thai et al., 2025), EA3D (Zhou et al., 2025), and EmbodiedSplat (Lee et al., 2026), there is a growing trend toward feed-forward architectures for 3DGS. Integrating such feed-forward mechanisms into X-GS may achieve faster speeds than the current optimization-based pipeline. However, such models also entail significant training costs for different VFMs. Finally, accommodating dynamic scenes using techniques such as 4DGS (Wu et al., 2024; Gao et al., 2025) represents a highly promising direction for embodied AI applications (Brohan et al., 2023) and world models (Maes et al., 2026).

Acknowledgements

The research presented in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 2300246, RGC C1043-24G), (CUHK 14203425, RGC GRF 2151317), and CUHK 7010870.

References

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. 2022. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, pages 19107–19117. IEEE.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, and 35 others. 2023. RT-2: vision-language-action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818.

Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 2022. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *CVPR*, pages 16443–16452. IEEE.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. 2025. Worldvla: Towards autoregressive action world model. *CoRR*, abs/2506.21539.

Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. 2020. Scanrefer: 3d object localization in RGB-D scans using natural language. In *ECCV (20)*, volume 12365 of *Lecture Notes in Computer Science*, pages 202–221. Springer.

Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. 2021. Scan2cap: Context-aware dense captioning in RGB-D scans. In *CVPR*, pages 3193–3203. Computer Vision Foundation / IEEE.

Hanlin Chen, Fangyin Wei, and Gim Hee Lee. 2024a. Chatsplat: 3d conversational gaussian splatting. *CoRR*, abs/2412.00734.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. 2024b. LL3DA: visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. In *CVPR*, pages 26418–26428. IEEE.

Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. 2023. End-to-end 3d dense captioning with vote2cap-detr. In *CVPR*, pages 11124–11133. IEEE.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. 2024a. Scene-llm: Extending language model for 3d visual understanding and reasoning. *CoRR*, abs/2403.11401.

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

Zhongpai Gao, Benjamin Planche, Meng Zheng, Anwesa Choudhuri, Terrence Chen, and Ziyan Wu. 2025. 7dgs: Unified spatial-temporal-angular gaussian splatting. *CoRR*, abs/2503.07946.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. 2024. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and RGB-D cameras. In *CVPR*, pages 21584–21593. IEEE.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: language embedded radiance fields. In *ICCV*, pages 19672–19682. IEEE.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. 2023. Segment anything. *CoRR*, abs/2304.02643.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing nerf for editing via feature field distillation. In *NeurIPS*.

Seungjun Lee, Zihan Wang, Yunsong Wang, and Gim Hee Lee. 2026. Embodiedsplat: Online feed-forward semantic 3dgs for open-vocabulary 3d scene understanding. *arXiv preprint arXiv:2603.04254*.

Sibaek Lee, Seongbo Ha, Kyeongsu Kang, Joonyeol Choi, Seungjun Tak, and Hyeonwoo Yu. 2025. LEGO-SLAM: language-embedded gaussian optimization SLAM. *CoRR*, abs/2511.16144.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, pages 7331–7341. Computer Vision Foundation / IEEE.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2025a. Llava-onevision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025.

Boyi Li, Kilian Q. Weinberger, Serge J. Belongie, Vladlen Koltun, and René Ranftl. 2022. Language-driven semantic segmentation. In *ICLR*. OpenReview.net.

Fei-Fei Li, Justin Johnson, Christoph Lassner, and Ben Mildenhall. 2024. What is spatial intelligence?

Haoyuan Li, Yanpeng Zhou, Tao Tang, Jifei Song, Yihan Zeng, Michael Kampffmeyer, Hang Xu, and Xiaodan Liang. 2025b. Unigs: Unified language-image-3d pretraining with gaussian splatting. In *ICLR*. OpenReview.net.

Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. 2025c. Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ FPS. *CoRR*, abs/2507.07136.

Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. 2024. Moe-llava: Mixture of experts for large vision-language models. *CoRR*, abs/2401.15947.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *CoRR*, abs/2304.08485.

Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric P. Xing, and Shijian Lu. 2023b. Weakly supervised 3d open-vocabulary segmentation. In *NeurIPS*.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. 2023. SQA3D: situated question answering in 3d scenes. In *ICLR*. OpenReview.net.

Yueen Ma, DaFeng Chi, Shiguang Wu, Yuecheng Liu, Yuzheng Zhuang, and Irwin King. 2025. Astra: Efficient transformer architecture and contrastive dynamics learning for embodied instruction following. In *EMNLP*.

Yueen Ma and Irwin King. 2025. 3d-moe: A mixture-of-experts multi-modal LLM for 3d vision and pose diffusion via rectified flow. *CoRR*, abs/2501.16698.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. 2026. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. *CoRR*, abs/2603.19312.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. 2024. Gaussian splatting SLAM. In *CVPR*, pages 18039–18048. IEEE.

Yuning Peng, Haiping Wang, Yuan Liu, Chenglu Wen, Zhen Dong, and Bisheng Yang. 2024. Gags: Granularity-aware 3d feature distillation for gaussian splatting. *arXiv preprint arXiv:2412.13654*.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. 2024. Langsplat: 3d language gaussian splatting. In *CVPR*, pages 20051–20060. IEEE.

Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. 2024. GOI: find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *ACM Multimedia*, pages 5328–5337. ACM.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. 2024. Language embedded 3d gaussians for open-vocabulary scene understanding. In *CVPR*, pages 5333–5343. IEEE.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, and 11 others. 2019. The replica dataset: A digital replica of indoor spaces. *CoRR*, abs/1906.05797.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, pages 573–580. IEEE.

Anh Thai, Songyou Peng, Kyle Genova, Leonidas J. Guibas, and Thomas A. Funkhouser. 2025. Splattalk: 3d VQA with gaussian splatting. *CoRR*, abs/2503.06271.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. 2025. VGGT: visual geometry grounded transformer. In *CVPR*, pages 5294–5306. Computer Vision Foundation / IEEE.

Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. 2024. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. In *NeurIPS*.

Tianci Wen, Zhiang Liu, and Yongchun Fang. 2025. Segs-slam: Structure-enhanced 3d gaussian splatting slam with appearance embedding. In *ICCV*, pages 28103–28113. IEEE.

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

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *CVPR*, pages 4389–4398. IEEE.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. 2024. GS-SLAM: dense visual SLAM with 3d gaussian splatting. In *CVPR*, pages 19595–19604. IEEE.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. 2022. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *ISMAR*, pages 499–507. IEEE.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. 2024. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV (29)*, Lecture Notes in Computer Science, pages 162–179. Springer.

Jisang Yoo, Gyeongjin Kang, Hyun-kyu Ko, Hyeonwoo Yu, and Eunbyung Park. 2025. Openmonogs-slam: Monocular gaussian splatting SLAM with open-set semantics. *CoRR*, abs/2512.08625.

Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. 2026. Infinitevggt: Visual geometry grounded transformer for endless streams. *CoRR*, abs/2601.02281.

Vladimir Yugay, Yue Li, Theo Gevers, and Martin R. Oswald. 2023. Gaussian-slam: Photo-realistic dense SLAM with gaussian splatting. *CoRR*, abs/2312.10070.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952. IEEE.

Chaoning Zhang, Dongshen Han, Sheng Zheng, Jinwoo Choi, Tae-Ho Kim, and Choong Seon Hong. 2023. Mobilesamv2: Faster segment anything to everything. *CoRR*, abs/2312.09579.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. 2024. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, pages 21676–21685. IEEE.

Xiaoyu Zhou, Jingqi Wang, Yuang Jia, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. 2025. EA3D: online open-world 3d object extraction from streaming videos. *CoRR*, abs/2510.25146.

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. 2022. NICE-SLAM: neural implicit scalable encoding for SLAM. In *CVPR*, pages 12776–12786. IEEE.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, pages 2899–2909. IEEE.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. 2025. Streaming 4d visual geometry transformer. *CoRR*, abs/2507.11539.

A 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents a 3D scene using a collection of anisotropic 3D Gaussians, operating as a point-based alternative to continuous neural implicit representations. Each Gaussian i in the comprehensive map parameter set Θ is explicitly defined by the following attributes:

- **Mean vector (Position):** $\mu_i \in \mathbb{R}^3$
- **Covariance matrix:** Σ_i
- **Opacity:** $\alpha_i \in [0, 1]$
- **Color features:** c_i (normally represented as spherical harmonics for view-dependence, though often simplified to base RGB values in dense SLAM configurations).

The unnormalized spatial distribution of a 3D Gaussian evaluated at a spatial point x is defined as:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right). \quad (5)$$

To maintain positive semi-definiteness during gradient-based optimization, the covariance matrix Σ_i is parameterized by a scaling matrix S_i and a rotation matrix R_i:

$$\Sigma_i = R_i S_i S_i^T R_i^T. \quad (6)$$

In order to render these primitives onto the image plane from a given camera pose, each 3D Gaussian is projected into a 2D elliptical splat. Given a learned viewing transformation W and the Jacobian J of the affine approximation of the projective alignment, the resulting 2D covariance matrix Σ_i' is calculated as:

$$\Sigma_i' = JW\Sigma_i W^T J^T. \quad (7)$$

This continuous state update sets a framework perfectly suited for dense tracking and mapping.

B 3DGS-based SLAM Objectives

Continuous optical depth models compute the expected color C along a camera ray by integrating transmittance $T(t)$, volume density $\sigma(t)$, and color $c(t)$. 3DGS discretizes this continuous integral using a point-based approximation. For a set of N depth-sorted 2D splats obtained by projecting 3D Gaussians onto the image plane given a camera pose $\mathbf{T} \in \mathrm{SE}(3)$, the color at pixel x evaluates to:

$$C(x \mid \mathbf{T}, \Theta) = \sum_{i=1}^{N} c_i \alpha_i' \prod_{j=1}^{i-1}(1-\alpha_j'), \quad (8)$$

where α_i' is the 2D splat opacity. Depth $D(x)$ is similarly rendered by substituting color c_i with the ray depth d_i.

Camera tracking is performed via Analysis-by-Synthesis. For an incoming frame at timestamp t, the pose is initialized using the online VFM's camera pose estimate. Because the 3DGS rasterizer is natively differentiable, precise tracking is achieved by freezing the map parameters Θ and running gradient descent on the camera extrinsic parameters to minimize the photometric and depth losses (if depth inputs are available):

$$\mathcal{L}_{\text{track}} = \mathcal{L}_{\text{photo}}(I_t, \hat{I}_t) + \lambda_d \|D_t - \hat{D}_t\|_1. \quad (9)$$

The global mapping thread optimizes the scene parameters Θ over a sliding window of keyframes $\mathcal{K}$ using the same combined loss, augmented with an isotropic regularization term $\mathcal{L}_{\text{iso}}$ to prevent heavily elongated artifacts:

$$\mathcal{L}_{\text{map}} = \sum_{k\in\mathcal{K}} \left(\mathcal{L}_{\text{photo}}(I_k, \hat{I}_k) + \lambda_d \|D_k - \hat{D}_k\|_1\right) + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}}. \quad (10)$$

C Online VQ Codebook Details

To stabilize the online EMA updates introduced in the main text, we implement active dead-code management and a warm-start initialization process.

Instead of starting from uniform counts, we initialize the EMA statistics using a confidence-filtered assignment pass over an initial buffer of semantic features. We transform the observed feature distances into confidence scores using a temperature parameter τ:

$$p_{nk} = \frac{\exp(-\|\mathbf{x}_n - \mathbf{e}_k\|_2/\tau)}{\sum_{j=1}^{L}\exp(-\|\mathbf{x}_n - \mathbf{e}_j\|_2/\tau)}. \quad (11)$$

Only samples for which the maximum confidence exceeds a threshold γ are allowed to contribute to the initialization of the counts N_k and sums $\mathbf{M}_k$:

$$m_n = \mathbb{I}\left[\max_k p_{nk} \geq \gamma\right]. \quad (12)$$

During the continuous online tracking stage, the confidence score masks unreliable incoming samples, while the actual codeword assignments use hard nearest-neighbor matching to ensure discrete latent separation.

Furthermore, to handle dead codes escaping the tracked field of view, we explicitly monitor the EMA accumulated mass. If a codeword's utilization falls below a dead-code threshold ($N_k < \delta$), it is immediately reinitialized:

$$\mathbf{M}_k \leftarrow \tilde{\mathbf{x}}, \qquad N_k \leftarrow 1 + \varepsilon, \tag{13}$$

where $\tilde{\mathbf{x}}$ is sampled directly from a small reservoir of recently observed semantic features.

D Grid-Sampled Semantic Supervision Details

This section details the mathematical formulation of our grid-sampled semantic supervision and the extraction of its targets.

Dynamic Grid Sampling. Let the full image domain be $\Omega = \{0, \dots, H-1\} \times \{0, \dots, W-1\}$. For a given stride $s \geq 1$ and spatial offset $\mathbf{o} = (o_h, o_w)$, we sample a sparse grid of coordinates:

$$(u, v) = (o_h + ms,\ o_w + ns), \tag{14}$$

for integer indices m, n, yielding a compact spatial resolution of $H_s \times W_s$. To prevent the optimized point cloud from overfitting to a static lattice, the offset $\mathbf{o}$ is sampled uniformly from $\{0, \dots, s-1\}^2$ at each optimization step. This ensures the sampled grid smoothly traverses all s^2 possible pixel phases over time.

Compact Predictions and Targets. Rather than forming a dense feature map, our custom rasterizer directly computes the compact prediction $\hat{\mathbf{G}} \in \mathbb{R}^{D \times H_s \times W_s}$, where $\hat{\mathbf{G}}_{:,m,n}$ corresponds to the rendered feature at (u, v). We construct a corresponding target map $\mathbf{G}^\star$ and binary validity mask $\mathbf{V} \in \{0,1\}^{1 \times H_s \times W_s}$ on this same grid, populated via one of two pathways depending on the semantic source:

1. Precomputed Region-Indexed Annotations: Given a discrete segmentation mask S (where -1 denotes background) and a region-feature table $\Phi \in \mathbb{R}^{R \times D}$, we query the table at each sampled location (u, v):

$$V_{m,n} = \mathbb{I}[S(u,v) \neq -1],$$
$$\mathbf{G}^\star_{:,m,n} = \begin{cases} \Phi_{S(u,v)}, & S(u,v) \neq -1, \\ \mathbf{0}, & S(u,v) = -1. \end{cases} \tag{15}$$

2. Online Continuous Feature Maps: For dense features streaming from an online VLM at a lower abstract resolution $\mathbf{P} \in \mathbb{R}^{D \times H_f \times W_f}$, we dynamically align the image coordinates to the low-resolution grid:

$$\tilde{u}_{m,n} = \mathrm{round}\left(u \cdot \frac{H_f - 1}{\max(H-1, 1)} \right),$$
$$\tilde{v}_{m,n} = \mathrm{round}\left(v \cdot \frac{W_f - 1}{\max(W-1, 1)} \right). \tag{16}$$

This yields the target $\mathbf{G}^\star_{:,m,n} = \mathbf{P}_{:,\tilde{u}_{m,n},\tilde{v}_{m,n}}$ with an unconstrained mask $V_{m,n} = 1$.

Masked Loss Evaluation. Semantic supervision is applied exclusively at valid sampled locations using a combination of cosine similarity and L1 distance:

$$\mathcal{L}_{\text{grid}} = \lambda_{\text{sem}} \Big[\mathcal{L}_{\cos}\Big(\mathbf{V} \odot \hat{\mathbf{G}},\ \mathbf{V} \odot \mathbf{G}^\star \Big) + \mathcal{L}_1\Big(\mathbf{V} \odot \hat{\mathbf{G}},\ \mathbf{V} \odot \mathbf{G}^\star \Big) \Big], \tag{17}$$

where $\odot$ denotes element-wise masking. The mask $\mathbf{V}$ serves a dual purpose: it filters out unannotated background regions (preventing Gaussians from learning empty features) and masks out padded, out-of-bounds coordinates that arise from maintaining rigid batched tensor shapes at large grid offsets.

E 3D Visual Grounding Details

Given a text prompt p, we encode it with OpenCLIP and normalize the resulting embedding:

$$\mathbf{t}^+ = \frac{E_{\text{text}}(p)}{\|E_{\text{text}}(p)\|_2}. \tag{18}$$

We also encode a small set of generic negative phrases

$$\mathcal{N} = \{\text{object},\ \text{things},\ \text{stuff},\ \text{texture}\}, \tag{19}$$

with normalized embeddings

$$\mathbf{t}^-_n = \frac{E_{\text{text}}(n)}{\|E_{\text{text}}(n)\|_2}, \qquad n \in \mathcal{N}. \tag{20}$$

As described in the main text, each Gaussian i is associated with mixture weights $\mathbf{w}_i \in \mathbb{R}^L$ over the shared codebook $\mathbf{E} \in \mathbb{R}^{L \times D}$. The decoded semantic feature $\hat{\mathbf{f}}_i \in \mathbb{R}^D$ of Gaussian i and its L_2-normalized counterpart $\tilde{\mathbf{f}}_i$ are given by:

$$\hat{\mathbf{f}}_i = \mathbf{E}^\top \mathbf{w}_i, \qquad \tilde{\mathbf{f}}_i = \frac{\hat{\mathbf{f}}_i}{\left\|\hat{\mathbf{f}}_i\right\|_2 + \varepsilon}. \tag{21}$$

Table 6: Summary of Benchmarking Specifications.

Dataset	# Scenes	Scenes Evaluated	# Queries
Online SLAM			
TUM RGB-D (Sturm et al., 2012)	3	`fr1_desk, fr2_xyz, fr3_office`	–
Replica (Straub et al., 2019)	8	`office0-4, room0-2`	–
3D Visual Grounding			
LERF (Kerr et al., 2023)	4	`teatime, figurines, ramen, waldo_kitchen`	66 (11–19/scene)
3D-OVS (Liu et al., 2023b)	5	`bed, bench, lawn, room, sofa`	31 prompts (155 pairs)
3D Vision-Language Reasoning			
Scan2Cap (Chen et al., 2021) (3D Captioning)	800	`ScanNet` (Validation set: 141)	51K
ScanQA (Azuma et al., 2022) (3D QA)	800	`ScanNet` (Validation set: 71)	41K
SQA3D (Ma et al., 2023) (Situated 3D QA)	650	`ScanNet` (Test set: 67)	33K

To measure how well Gaussian i matches the prompt, we contrast the positive prompt embedding against each negative phrase:

$$r_i(p) = \min_{n \in \mathcal{N}} \text{softmax}\left(\tau \begin{bmatrix} \tilde{\mathbf{f}}_i^\top \mathbf{t}^+ \\ \tilde{\mathbf{f}}_i^\top \mathbf{t}_n^- \end{bmatrix}\right)_1, \tag{22}$$

where $(\cdot)_1$ denotes the probability assigned to the positive prompt, and $\tau = 10$ in our implementation. Intuitively, a Gaussian receives a high relevance score only if its decoded feature is more aligned with the prompt than with the generic negative phrases.

A prompt-specific Gaussian mask is then obtained by thresholding:

$$m_i(p) = \mathbb{I}[r_i(p) > \delta]\,. \tag{23}$$

For pixel-level prompt rendering, the renderer first splats the Gaussian mixture weights onto the image plane. Let $\mathbf{w}(x) \in \mathbb{R}^L$ denote the rendered mixture weights at pixel x. The semantic feature at that pixel is reconstructed and normalized as:

$$F(x) = \mathbf{E}^\top \mathbf{w}(x), \qquad \tilde{F}(x) = \frac{F(x)}{\|F(x)\|_2 + \varepsilon}. \tag{24}$$

Its prompt relevance is computed analogously:

$$R(x \mid p) = \min_{n \in \mathcal{N}} \text{softmax}\left(\tau \begin{bmatrix} \tilde{F}(x)^\top \mathbf{t}^+ \\ \tilde{F}(x)^\top \mathbf{t}_n^- \end{bmatrix}\right)_1. \tag{25}$$

The resulting relevance map is used to highlight image regions that are most semantically aligned with the text prompt.

F Entropy-Adaptive Gaussian Sampling for Generative VLMs

As demonstrated by recent works such as SplatTalk (Thai et al., 2025), projecting 3D Gaussian features into discrete tokens enables Large Language Models (LLMs) to perform zero-shot 3D visual question answering and scene understanding. However, directly passing all N_g Gaussians from an uncompressed SLAM map into the vision tower of a Generative VLM is computationally prohibitive and yields highly redundant tokens.

Rather than relying on naive spatial downsampling, we condense the scene into a compact sequence of M informative tokens ($M \ll N_g$) by ranking Gaussians according to the feature-channel entropy of their VQ-decoded semantic embeddings. This entropy is computed after VQ decoding and is not the entropy of the Perceiver's codeword-assignment weights.

Let $\hat{\mathbf{f}}_i \in \mathbb{R}^D$ denote the decoded semantic embedding of Gaussian i. For ranking, we apply a softmax across its D feature channels to obtain a normalized channel distribution $q_{i,d}$:

$$q_{i,d} = \frac{\exp(\hat{f}_{i,d})}{\sum_{j=1}^{D} \exp(\hat{f}_{i,j})}. \tag{26}$$

The feature-channel entropy H_i is then calculated as

$$H_i = -\sum_{d=1}^{D} q_{i,d} \log q_{i,d}. \tag{27}$$

Low entropy indicates that the normalized response is concentrated in a small number of embedding channels, whereas high entropy indicates that it is distributed across more channels. We use high feature-channel entropy as a heuristic for selecting feature-rich Gaussian tokens.

To select the final M tokens, we rank the Gaussians by these entropy scores and extract the subset $\mathcal{S}$ with the highest feature-channel entropy:

$$\mathcal{S} = \underset{i \in \{1,\ldots,N_g\}}{\arg\text{top-}M}\, H_i. \tag{28}$$

The resulting M Gaussians are then projected into the VLM's embedding space. This determinis-

tic sampling reduces the token count while prioritizing decoded embeddings with high feature-channel entropy. The resulting token sequence bridges the continuous geometric scene representation with the VLM's discrete context window for tasks such as 3D visual question answering (VQA) and zero-shot scene captioning.

G Benchmark Details

We provide benchmark details in Table 6.

H Additional Qualitative Results

We provide an additional qualitative example demonstrating the 3D scene captioning capabilities of X-GS in Figure 5.

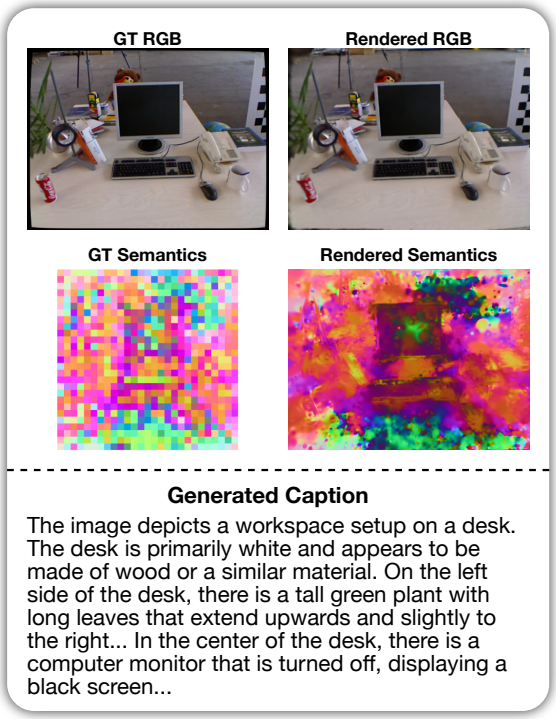


Figure 5: Qualitative results of X-GS for 3D scene caption generation. The VLM generates captions in a zero-shot setting by taking 3D semantic Gaussians as input. Note that only a portion of the generated content is shown.

I Towards Embodied AI and World Models

Finally, as a future direction for embodied AI applications, the X-GS-*Thinker* can be configured to interface with a Vision-Language-Action (VLA) model (Brohan et al., 2023; Ma et al., 2025; Ma and King, 2025). By feeding language-aligned 3D geometric features directly into a VLA, our system provides the spatial information required to support embodied tasks.

Additionally, X-GS serves as a natural foundation for developing 3D-centric world models (Maes et al., 2026). Existing world models typically predict future states via computationally expensive 2D image or video generation (Cen et al., 2025). In contrast, a 3DGS-based world model operates with significantly greater efficiency; simulating an action such as object manipulation simply requires segmenting the relevant Gaussians and applying rigid affine transformations (e.g., translation and rotation)—an inherently lightweight computational operation (Xie et al., 2024).