CapFrame: Text-Instructed Viewpoint Grounding in 3D Gaussian Scenes via Geometric Pseudo Labels

Jirong Li[1], Satoshi Ikehata[2,3], Shuhei Kurita[3], and Ikuro Sato[1,2]

[1] Institute of Science Tokyo, Japan
[2] Denso IT Laboratory, Inc., Japan
[3] National Institute of Informatics, Japan
jirong_li@d-itlab.c.titech.ac.jp

Abstract. 3D Gaussian Splatting (3DGS) enables photorealistic real-time novel view synthesis, yet placing a virtual camera to capture a desired frame remains largely manual. Existing language-guided approaches in 3D scenes mainly focus on *object-centric grounding*, determining *what* to observe but rarely controlling *how* it should appear in a single frame, such as subject orientation or frame layout. To address this limitation, we introduce a new task, Text-Instructed Viewpoint Grounding (TIVG), which aims to identify a 6-DoF camera pose in a 3D Gaussian scene whose rendered frame aligns with a text instruction. To solve this task, we propose *CapFrame*, a partially differentiable framework that converts language into *geometric pseudo labels* for camera pose optimization. CapFrame follows a *Retrieve–Translate–Refine* pipeline: it retrieves relevant views and ranks them through a *Question-Evaluation* process with MLLMs, translates the instruction into orientation and layout pseudo labels, and refines the camera pose via differentiable optimization with layout and orientation losses in 3DGS. Experiments on 38 real-world scenes with 135 instructions indicate that CapFrame produces viewpoints better aligned with texts than heuristic viewpoint search and adapted trajectory generation baselines, validated by VLM metrics, MLLM judges, and user studies. Code is available at: https://github.com/jirongli/CapFrame

Keywords: 3D Gaussian Splatting · Viewpoint Grounding · Camera Pose Optimization · MLLM

1 Introduction

Recent advances in 3D representation technologies such as Neural Radiance Fields (NeRF) [38] and 3D Gaussian Splatting (3DGS) [19], enable photorealistic novel view synthesis for applications spanning VR/AR [16, 63], content creation [62,67], and scene editing [4,54]. Notably, 3DGS has gained prominence due to its real-time rendering and high visual fidelity, making it well-suited for interactive environments. However, despite its efficient rendering, identifying desirable viewpoints still requires tedious manual trial-and-error and scales poorly

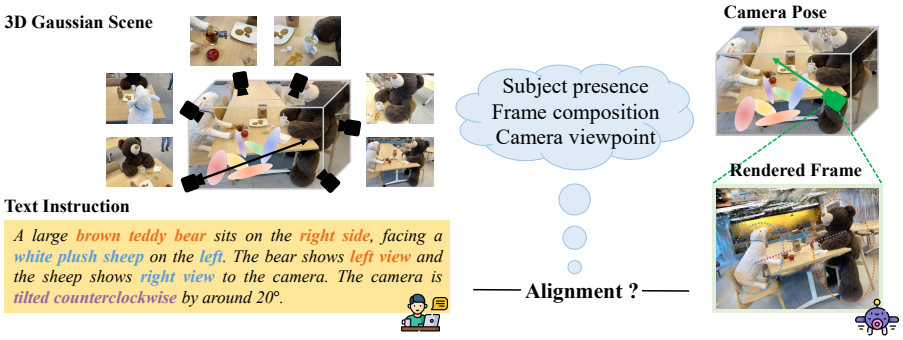

Fig. 1: Text-instructed viewpoint grounding in a 3DGS scene. Given a 3D Gaussian scene and a text instruction, the goal is to identify a camera pose whose rendered frame aligns with the described subject, composition and viewpoint.

as scenes grow larger and more complex. Therefore, intent-aligned automatic viewpoint recommendation is crucial for intuitive scene interaction.

Recent language-guided 3D methods [27, 42, 47, 57] excel at semantic object localization and trajectory generation [8, 28, 33, 34, 65], but primarily focus on *object-centric grounding*. They identify "what" to observe but overlook "how" it should be presented. Real-world instructions are inherently compositional, specifying orientation, spatial relationships, or photographic attributes (*e.g.*, "*a close-up photo of a cat from the front*"). Existing work ensures object presence but fails to enforce such intentional framing constraints. While Multimodal Large Language Models (MLLMs) [1, 13, 31, 60] offer strong zero-shot spatial reasoning to interpret such constraints, translating instruction text into concrete 3DGS viewpoints remains unexplored.

In this work, we formalize a new task, *Text-Instructed Viewpoint Grounding* for 3DGS, which aims to identify a 6-DoF camera pose that captures a frame aligned with a text instruction. To accomplish this, we propose **CapFrame**, a partially differentiable framework for text-instructed camera placement in 3D Gaussian scenes. CapFrame follows a three-stage **Retrieve**–**Translate**–**Refine** pipeline. In the **Retrieve** stage, we retrieve semantically relevant views from the training images and perform fine-grained ranking through a *Question-Evaluation* (QE) process guided by an MLLM, providing initialization for camera pose optimization. In the **Translate** stage, we convert compositional language into geometric pseudo labels, including subject–Gaussian associations, orientation pseudo labels, and layout pseudo labels. In the **Refine** stage, exploiting the differentiability of 3DGS, we optimize the camera pose by backpropagating layout and orientation losses directly to the pose.

Extensive experiments show that CapFrame effectively grounds compositional text instructions to camera poses in diverse 3D Gaussian scenes. Across 38 real-world scenes and 135 curated instructions, it consistently produces viewpoints whose rendered frames better satisfy compositional requirements than

heuristic viewpoint search baselines and adapted trajectory generation methods. Quantitative evaluation with text-image similarity metrics, MLLM judges, and a user study further confirms stronger alignment between rendered frames and textual descriptions.

2 Related Work

3D Gaussian Representation. 3D scene representations range from implicit radiance fields [38, 40, 48] to explicit structures such as meshes [17, 18], point clouds [26, 41], and voxels [46, 50]. Addressing the rendering latency of implicit models and the topological rigidity of explicit ones, 3D Gaussian Splatting (3DGS) [19] represents scenes as anisotropic Gaussian primitives. With real-time rasterization and high visual fidelity, 3DGS has become a popular representation for novel view synthesis. Subsequent work applies 3DGS to tasks including 3D segmentation [6, 15, 61], scene editing [4, 54, 56], generative content creation [5, 62, 67], and dynamic scene modeling [36, 55]. Systems such as MonoGS [37] further show that differentiable rasterization in 3DGS enables robust 6-DoF pose optimization. Building on this property, we exploit differentiable 3DGS rendering to optimize camera viewpoints aligned to text instruction.

Vision-Language Understanding. Vision-Language Models (VLMs) learn joint visual-textual embeddings from image-text pairs. CLIP [43] established strong contrastive alignment, followed by models improving multimodal understanding [9, 24, 25, 51, 58, 64]. To extend such semantics to 3D scenes, recent work [27, 35, 42, 47, 57] distills language features into 3D Gaussian primitives, enabling open-vocabulary querying and amodal reasoning under occlusion. However, these approaches remain largely object-centric, focusing on *what* to attend to rather than *how* to frame it. Meanwhile, Multimodal Large Language Models (MLLMs) [1, 13, 31, 60] exhibit strong zero-shot spatial reasoning, enabling tasks like object reasoning [59] and physical simulation [66]. We study viewpoint alignment conditioned on compositional language, using MLLMs to convert photographic instructions into geometric pseudo supervisions for camera viewpoint optimization.

Camera Control. Camera control in virtual environments aims to satisfy cinematic and narrative constraints. Early methods relied on mathematical formulations [3, 11, 30] or rule-based systems encoding cinematic heuristics [10, 12]. Recent learning-based approaches [8, 28, 33, 65] generate language-conditioned camera motion from large-scale data. For example, ChatCam [33] enables conversational camera navigation, while GenDoP [65] synthesizes cinematic trajectories. Optimization-based methods such as JAWS [52] and SplaTraj [34] refine camera paths directly in 3D representations via visibility objectives. In particular, SplaTraj [34] combines 3DGS with continuous language fields [42] to maintain visibility of open-vocabulary targets. However, these methods do not address fine-grained compositional constraints for intentional photographic framing.

3 Preliminaries: 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [19] represents a scene as anisotropic Gaussians $\mathcal{G}$, each with mean $\mu_W \in \mathbb{R}^3$ and covariance $\Sigma_W \in \mathbb{R}^{3\times3}$:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu_W)^\top \Sigma_W^{-1}(x-\mu_W)\right). \tag{1}$$

We parameterize Σ_W as $\Sigma_W = RSS^\top R^\top$ to ensure validity.

Let $\boldsymbol{T}_{CW} \in SE(3)$ be the 6-DoF camera pose. During rendering, each Gaussian is projected to the image plane with

$$\mu_I = \pi(\boldsymbol{T}_{CW}\mu_W), \qquad \Sigma_I = J\boldsymbol{R}\Sigma_W\boldsymbol{R}^\top J^\top, \tag{2}$$

where $\pi(\cdot)$ is perspective projection, $\boldsymbol{R}$ is rotation of $\boldsymbol{T}_{CW}$, and J is projection Jacobian. Pixel color is obtained by alpha compositing $\mathcal{N}$ overlapping Gaussians:

$$C = \sum_{k\in\mathcal{N}} c_k\alpha_k \prod_{j=1}^{k-1}(1-\alpha_j), \tag{3}$$

with c_k and α_k the color and opacity of the k-th Gaussian.

Since rasterization is differentiable, gradients propagate to $\boldsymbol{T}_{CW}$. Following MonoGS [37], for a pose-dependent function f, we define the manifold derivative using $\tau \in \mathfrak{se}(3)$:

$$\frac{Df(\boldsymbol{T}_{CW})}{D\boldsymbol{T}_{CW}} = \lim_{\tau\to0} \frac{\log\big(f(\exp(\tau)\circ\boldsymbol{T}_{CW})\circ f(\boldsymbol{T}_{CW})^{-1}\big)}{\tau}. \tag{4}$$

This enables gradient-based optimization of 6-DoF camera poses.

4 Text-Instructed Viewpoint Grounding in 3DGS

We formalize the task of *Text-Instructed Viewpoint Grounding* (TIVG) within a 3D Gaussian scene. Let $\mathcal{S} = \{G_i\}_{i=1}^N$ denote a 3DGS scene reconstructed from a set of training images $\mathcal{C}$, and let T be a natural language instruction describing a desired viewpoint. A camera pose, comprising a rotation matrix $\boldsymbol{R} \in SO(3)$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$, determines the differentiable rasterization $\mathcal{R}$ of the scene:

$$\boldsymbol{I} = \mathcal{R}(\mathcal{S}, \boldsymbol{T}_{CW}), \quad \text{where} \quad \boldsymbol{T}_{CW} = \begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0}^\top & 1\end{bmatrix} \in SE(3). \tag{5}$$

The objective is to identify a camera pose $\boldsymbol{T}_{CW}$ that produces a rendered image $\boldsymbol{I} \in \mathbb{R}^{3\times H\times W}$ aligned with the instruction T. For instance, as illustrated in Fig. 1, given the instruction "*A large brown teddy bear sits on the right side, showing left view to the camera*", the system should not only localize the bear but also determine a precise 6-DoF configuration that satisfies both the orientation (left view) and the layout (right side of the frame).

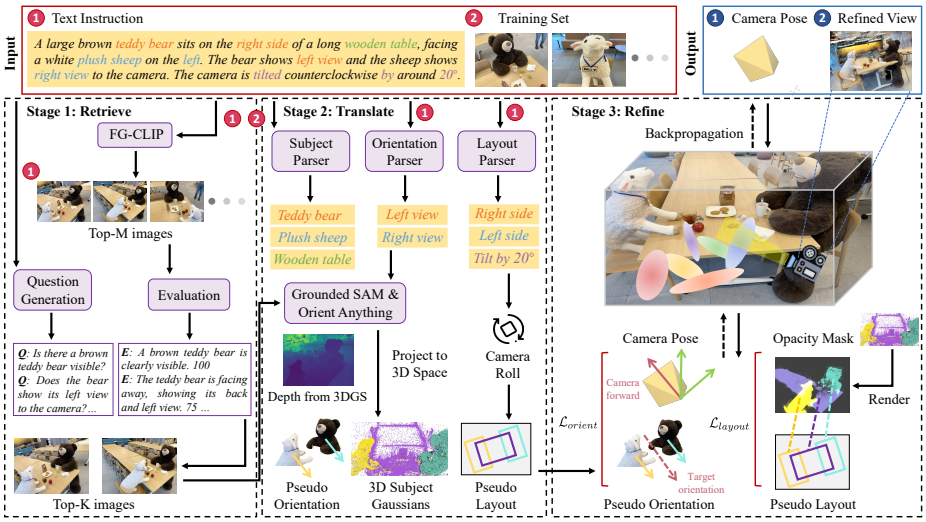

Fig. 2: Overview of CapFrame. Starting from compositional text and views in training set, we retrieve images consistent with the text. A Question-Evaluation process within the MLLM is introduced to perform fine-grained ranking, selecting relevant image subset. Subsequently, the MLLM translates the text into geometric regularizers. In conjunction with pretrained models, we construct geometric pseudo labels based on the image subset for orientation and layout. Finally, guided by these pseudo labels, we refine camera poses through differentiable optimization in the 3D Gaussian scene.

Unlike object-centric localization [42, 57], TIVG does not generally admit a unique solution, as multiple viewpoints may satisfy the same compositional description. Given this inherent non-uniqueness, we evaluate this task using text-image similarity scores computed by external VLMs, alignment ratings from two MLLM judges, together with perceptual user studies, as detailed in Sec. 6.

5 Method

We introduce **CapFrame** to bridge the gap between abstract instructions and precise 6-DoF poses via a three-stage pipeline (Fig. 2): **(1) Retrieve:** We identify and rank semantic anchor views from training images $\mathcal{C}$ via a *Question-Evaluation* (QE) process using an MLLM (Sec. 5.1). **(2) Translate:** We convert linguistic constraints into geometric pseudo labels for orientation and layout (Sec. 5.2). **(3) Refine:** Initialized by retrieved poses, we optimize the camera by backpropagating layout and orientation losses to pose parameters (Sec. 5.3).

5.1 Retrieve: Semantic-aware Viewpoint Initialization

Exhaustively searching the continuous $SE(3)$ space of a 3DGS scene is computationally prohibitive. Moreover, gradient-based camera pose optimization requires

a suitable initialization to ensure stable convergence. Since 3DGS scenes are reconstructed from a finite set of training images $\mathcal{C}$ that provide near-complete scene coverage, we assume that these existing views form a suitable discrete subspace for initializing the optimization. Therefore, in this **Retrieve** stage, we aim to identify the image from $\mathcal{C}$ most relevant to the input text instruction.

Given a text instruction T, we first perform global semantic alignment using FG-CLIP [58] to identify a relevant subset of candidate poses. Since reasoning with MLLMs later is computationally expensive, this lightweight filtering step significantly reduces the candidate space. We extract a text embedding f_T and image embeddings $f_{I,i}$ for each view $I_i \in \mathcal{C}$. Similarity is computed via cosine similarity $\langle f_T, f_{I,i} \rangle$, and the top-$M$ view-pose pairs are retained as

$$\mathcal{C}_g = \{(I_i, \boldsymbol{T}_{CW,i})\}_{i=1}^{M}, \tag{6}$$

where $M \approx |\mathcal{C}|/8$. While efficient, our preliminary experiments reveal that simple VLM-based global alignment often fails in cluttered scenes with multiple objects, as it struggles to discriminate nuanced compositional requirements.

To overcome the limitations of holistic VLM embeddings, we introduce a **Question-Evaluation (QE)** strategy that leverages the reasoning of multimodal large language models (MLLMs), such as Qwen3-VL [1]. Instead of relying on a single similarity score, QE decomposes T into a set of questions $\mathcal{Q}$ targeting specific semantic and photographic attributes (*e.g.*, "*Is the subject visible?*" or "*Is the subject located on the right side of the frame?*"). These questions are automatically generated by MLLM using a fixed, task-agnostic prompt template, ensuring a consistent and hands-free evaluation process across diverse scenes.

The MLLM then evaluates each view $I_i \in \mathcal{C}_g$ against $\mathcal{Q}$ to produce granular scores. By averaging these scores, we obtain a ranking that reflects complex compositional alignment. The top-K view-pose pairs are retained as

$$\mathcal{C}_f = \{(I_i, \boldsymbol{T}_{CW,i})\}_{i=1}^{K}. \tag{7}$$

This reasoning-based filtering provides a high-quality initialization for the subsequent optimization stage, ensuring that the starting viewpoint already satisfies basic semantic constraints.

5.2 Translate: From Language to Geometric Pseudo Labels

The **Retrieve** stage provides a strong initialization by selecting candidate viewpoints from the training set. Our goal, however, is to search the *continuous* camera pose space in $SE(3)$ and find a viewpoint whose rendered frame matches the text instruction. Because natural language rarely specifies optimization-ready numeric targets (*e.g.*, exact angles or pixel coordinates), directly constructing differentiable objectives from text is challenging. We therefore introduce a translation step that converts compositional instructions into *geometric pseudo labels*—structured targets guiding continuous pose refinement in 3DGS.

We follow key decisions in photographic composition: selecting salient subjects, defining viewing direction, and arranging subjects within the frame. Accordingly, we extract subject Gaussians and construct two pseudo-label types:

(i) orientation pseudo labels for viewpoint control and (ii) layout pseudo labels for framing constraints.

Subject–Gaussian Association. Let $\mathcal{C}_f = \{(I_j, \boldsymbol{T}_{CW,j})\}_{j=1}^{K}$ denote the top-K retrieved view-pose pairs. Given instruction T, we prompt an MLLM to extract key subjects $\mathcal{O} = \{O_1, \ldots, O_n\}$, including mentioned entities and visually dominant objects affecting framing. Each subject is associated with a subset of 3D Gaussians to enable subject-centric reasoning and differentiable mask construction. Unlike open-vocabulary localization methods that embed language features into all Gaussians [42, 57], we localize only the small set $\mathcal{O}$, which suffices for constructing pseudo labels.

For each subject O_i and retrieved view I_j, we obtain a segmentation mask M_{ij}^{SAM} using Grounded SAM [22,32,44]. Pixels $(u, v) \in M_{ij}^{\text{SAM}}$ are back-projected using depth rendered from 3DGS to obtain a 3D point set

$$\mathcal{P}_{ij} = \{(X_W, Y_W, Z_W)^\top \mid (u, v) \in M_{ij}^{\text{SAM}}\}. \tag{8}$$

Since rendered depth may be noisy, these points may not coincide exactly with Gaussian means. We therefore retrieve a subject-specific Gaussian subset $\mathcal{G}_{Q,i} \subset \mathcal{G}$ via KNN search between $\cup_j \mathcal{P}_{ij}$ and Gaussian means $\{\mu_{W,k}\}$. The subject centroid is computed as

$$\boldsymbol{C}_{W,i} = \frac{1}{|\mathcal{G}_{Q,i}|} \sum_{k \in \mathcal{G}_{Q,i}} \mu_{W,k}. \tag{9}$$

Rendering only $\mathcal{G}_{Q,i}$ yields a differentiable subject mask for layout objectives.

Orientation Pseudo Labels. Many instructions specify viewpoint and subject orientation through cues such as *front view*, *side view*, or *look down*. We convert such language into orientation pseudo labels by combining (i) relative offsets inferred from *text* and (ii) estimated subject orientation from images.

Given T and $\mathcal{O}$, we prompt the MLLM to infer relative orientation offsets $\Delta\boldsymbol{d}_i = (\Delta\phi_i, \Delta\theta_i, \Delta\gamma_i)$, representing azimuth, elevation and roll adjustments. Orientation constraints are applied only to asymmetric subjects with meaningful front or back semantics. Symmetric objects (*e.g.*, balls) rely solely on layout constraints (subject types are identified by MLLM).

To estimate current orientation of *subjects*, we apply Orient Anything [53] to the top-K images, obtaining $(\phi_{O,i}, \theta_{O,i})$ ($\gamma_{O,i}$ is not used as we only need the forward direction of the subject). The target orientation combines this estimate with the text-derived offsets:

$$\boldsymbol{P}_{O,i}^{\text{orient}} = (\phi_{O,i} + \Delta\phi_i,\ \theta_{O,i} + \Delta\theta_i,\ \Delta\gamma_i), \tag{10}$$

where azimuth is wrapped modulo and elevation clamped to a valid range. The result is *mapped to world coordinates*, producing $\boldsymbol{P}_{W,i}^{\text{orient}}$ used as a differentiable regularizer during refinement.

Layout Pseudo Labels. Photographic intent also specifies subject placement within the frame (*e.g.*, *subject on the right*, *centered*, *close-up*). We therefore prompt the MLLM to infer a target 2D layout in normalized image coordinates. For each subject O_i, the layout pseudo label is a bounding box

$$\boldsymbol{P}_i^{\text{layout}} = [x_{\min}, y_{\min}, x_{\max}, y_{\max}], \tag{11}$$

which is converted to pixel coordinates using image width W and height H.

To remain consistent with roll constraints, we adjust the layout if $\Delta\gamma_i \neq 0$. Let $\boldsymbol{c} = (c_x, c_y)$ denote the center of the bounding box and $\boldsymbol{b}$ denote a corner of the bounding box. Each corner is rotated around $\boldsymbol{c}$ by $\Delta\gamma_i$:

$$\boldsymbol{b}_\gamma = \boldsymbol{c} + \boldsymbol{R}(\Delta\gamma_i)(\boldsymbol{b} - \boldsymbol{c}), \tag{12}$$

where $\boldsymbol{R}(\Delta\gamma_i)$ is the 2D rotation matrix. The rotated corners define the final layout target used during refinement.

5.3 Refine: Gradient-Based Camera Pose Optimization

Starting from the pose retrieved in Sec. 5.1, we optimize an incremental update $\tau = [\Delta\boldsymbol{r}, \Delta\boldsymbol{t}]^\top \in \mathfrak{se}(3)$ and update the camera pose by left composition $\boldsymbol{T}_{CW} \leftarrow \exp(\tau) \circ \boldsymbol{T}_{CW}$. We minimize a multi-objective loss

$$\mathcal{L} = \mathcal{L}_{\text{layout}} + \mathcal{L}_{\text{orient}}, \tag{13}$$

where both terms are derived from pseudo labels in Sec. 5.2.

Layout Loss. The layout loss enforces composition constraints by encouraging each subject to occupy its target region $\boldsymbol{P}_i^{\text{layout}}$ (Sec. 5.2). For subject O_i, we render only its Gaussian subset $\mathcal{G}_{Q,i}$ and obtain a differentiable opacity mask $M_i^g \in [0,1]^{H\times W}$ via alpha compositing:

$$M_i^g(u,v) = \sum_{k=1}^{N_i} \alpha_k(u,v) \prod_{j<k} \big(1 - \alpha_j(u,v)\big), \tag{14}$$

where N_i is the number of contributing Gaussians and $\alpha_k(u,v)$ denotes the opacity of the k-th Gaussian. Let $\boldsymbol{C}_{P,i}$ denote the soft centroid of M_i^g:

$$\boldsymbol{C}_{P,i} = \frac{\sum_{(u,v)\in\Omega}(u,v)\, M_i^g(u,v)}{\sum_{(u,v)\in\Omega} M_i^g(u,v) + \epsilon}, \tag{15}$$

where Ω is the image domain. If $\boldsymbol{C}_{P,i}$ lies inside $\boldsymbol{P}_i^{\text{layout}}$, the loss is zero; otherwise we penalize the distance to the nearest point on the region boundary $\partial P_i^{\text{layout}}$:

$$\mathcal{L}_{\text{center}}^i = \begin{cases} 0, & \text{if } \boldsymbol{C}_{P,i} \in \boldsymbol{P}_i^{\text{layout}}, \\ \min_{\boldsymbol{p}\in\partial P_i^{\text{layout}}} \left\|\boldsymbol{C}_{P,i} - \boldsymbol{p}\right\|_2, & \text{otherwise.} \end{cases} \tag{16}$$

Empirically, centroid constraints alone are insufficient when subjects are elongated or partially outside the target region. We therefore additionally encourage mask coverage inside the box and penalize leakage outside it. Using the indicator $\mathbb{I}(\cdot)$, we define

$$\begin{aligned}\mathcal{L}^i_{\text{in}} &= 1 - \frac{\sum_{(u,v)\in\Omega} M^g_i(u,v)\,\mathbb{I}\big((u,v)\in \boldsymbol{P}^{\text{layout}}_i\big)}{\sum_{(u,v)\in\Omega} M^g_i(u,v)+\epsilon},\\ \mathcal{L}^i_{\text{out}} &= \frac{\sum_{(u,v)\in\Omega} M^g_i(u,v)\,\mathbb{I}\big((u,v)\notin \boldsymbol{P}^{\text{layout}}_i\big)}{\sum_{(u,v)\in\Omega} M^g_i(u,v)+\epsilon}.\end{aligned} \tag{17}$$

The total layout loss aggregates all subjects:

$$\mathcal{L}_{\text{layout}} = \sum_{i=1}^{n} \left(\lambda_c \mathcal{L}^i_{\text{center}} + \lambda_{\text{in}} \mathcal{L}^i_{\text{in}} + \lambda_{\text{out}} \mathcal{L}^i_{\text{out}}\right), \tag{18}$$

where λ_c, λ_{in}, and λ_{out} are scalar weights and ϵ ensures numerical stability.

Orientation Loss. The orientation loss enforces viewpoint constraints from orientation pseudo labels (Sec. 5.2). It contains three terms: gravity (discouraging tilt), forward-facing (aligning the camera with the desired viewpoint), and look-at (keeping the subject visible).

Gravity. When roll is not specified (*i.e.*, $\Delta\gamma_i = 0$), we encourage an upright camera by aligning the camera up direction $\boldsymbol{D}_{C,cu}$ with the world up direction in the camera frame $\boldsymbol{D}_{C,wu}$:

$$\mathcal{L}_{\text{gravity}} = 1 - \langle \boldsymbol{D}_{C,cu}, \boldsymbol{D}_{C,wu} \rangle. \tag{19}$$

If roll is specified, this term is disabled and roll is handled by the roll-corrected layout labels (Sec. 5.2).

Forward-facing. For asymmetric subjects $O_i \in \mathcal{O}_w$, we align the camera forward direction $\boldsymbol{D}_{C,cf}$ with the target orientation $\boldsymbol{P}^{\text{orient}}_{C,i}$:

$$\mathcal{L}^i_{\text{forward}} = 1 - \langle \boldsymbol{D}_{C,cf}, \boldsymbol{P}^{\text{orient}}_{C,i} \rangle. \tag{20}$$

Look-at. To keep the subject in view, we align the camera with the subject centroid. Let $\boldsymbol{C}_{cam}$ and $\boldsymbol{C}_{W,i}$ denote the camera center and subject centroid (Sec. 5.2). The normalized direction

$$\boldsymbol{d}_{W,i} = \frac{\boldsymbol{C}_{W,i} - \boldsymbol{C}_{cam}}{\|\boldsymbol{C}_{W,i} - \boldsymbol{C}_{cam}\|_2} \tag{21}$$

is expressed in the camera frame as $\boldsymbol{D}_{C,cam,i}$, yielding

$$\mathcal{L}^i_{\text{look-at}} = 1 - \langle \boldsymbol{D}_{C,cf}, \boldsymbol{D}_{C,cam,i} \rangle. \tag{22}$$

The final orientation loss is

$$\mathcal{L}_{\text{orient}} = \lambda_g \mathcal{L}_{\text{gravity}} + \sum_{O_i \in \mathcal{O}_w} \left(\lambda_f \mathcal{L}^i_{\text{forward}} + \lambda_l \mathcal{L}^i_{\text{look-at}}\right), \tag{23}$$

where λ_g, λ_f, and λ_l are scalar weights.

6 Results

Implementation Details. We use Qwen3-VL [1] as the MLLM for text reasoning in the **Retrieve** stage and pseudo-label generation in the **Translate** stage. For scene representation, we adopt the standard 3DGS implementation [19] with camera pose gradients from MonoGS [37] (Eq. (4)). As our focus is viewpoint grounding, the reconstruction method is not critical. In **Retrieve**, the number of top-ranked views for fine-grained ranking is set to $K = 2$. Camera pose refinement in **Refine** is optimized using Adam [21] with learning rates 0.007 (rotation) and 0.005 (translation). We set $\lambda_c = 1$ and $\lambda_{\text{in}} = 6$, while all other weights are 2 ($\lambda_{\text{out}}, \lambda_g, \lambda_f, \lambda_l$). Optimization runs for up to 1500 iterations with early stopping when the loss change falls below 1×10^{-4} or 10^{-5} depending on scene scale. Experiments run on a single NVIDIA H100 GPU (80GB).

Baselines. As no established baselines exist for text-instructed viewpoint grounding in 3DGS scenes, we construct two heuristic baselines inspired by scene exploration methods [49]: *Interpolation-based Viewpoint Search* (IVS) and *Sampling-based Viewpoint Search* (SVS). IVS generates candidate viewpoints by interpolating between top-K retrieved camera poses from the **Retrieve** stage (Sec. 5.1). SVS instead samples camera poses densely on a sphere centered around the key subjects identified in the **Translate** stage (Sec. 5.2). For both baselines, each candidate pose is rendered using 3DGS, and the final viewpoint is selected as the frame with the highest text-image similarity measured by FG-CLIP [58]. In addition, we adapt the relevant components of two trajectory generation methods, ChatCam [33] and SplaTraj [34], to our TIVG setting. We build on ChatCam's available *Anchor Determination*, which optimizes camera poses via CLIP [43] gradients, and reproduce the relevant part of SplaTraj.

Datasets. We evaluate on real-world 3D reconstruction datasets: 5 scenes from *Mip-NeRF 360* [2], 13 from *Deep Blending* [14], 4 from *Tanks and Temples* [23], 4 from *LERF-OVS* [20], and 12 from *DL3DV-10K* [29], totaling 38 indoor and outdoor scenes. For scenes without camera poses, we estimate them by COLMAP [45]. We manually curate 3-4 instructions per scene, resulting in 135 text descriptions.

Metrics. As text-instructed viewpoint grounding lacks a unique ground-truth viewpoint, evaluation is non-trivial. For quantitative evaluation, we compute text-image similarity using CLIP [43] (ViT-H/14) and SigLIP2 [51]. Using two VLMs reduces bias from a single scoring function and provides a more robust estimate of semantic alignment. Similarly, we introduce two MLLM judges (GPT-5.4-mini [39] and Gemini-2.5-Flash [7]) as blind photographic evaluators that rate the compositional *alignment scores* (AS) and select the most aligned output frame across methods, computing the *win rates* (WR) of CapFrame over the baselines. We additionally conduct a user study to evaluate perceptual alignment. The study involves 33 participants, each completing questionnaires with 12 groups. In each group, participants rate three candidate frames on a 5-point

Fig. 3: Text-instructed viewpoint grounding in diverse 3DGS scenes. Given a natural language instruction (left), CapFrame identifies a camera pose that produces a frame consistent with the described subject, composition and viewpoint. Starting from retrieved initial views, the camera pose is refined through differentiable optimization in the 3D Gaussian scene to generate the final rendered frame (middle), with the optimized camera pose shown on the right.

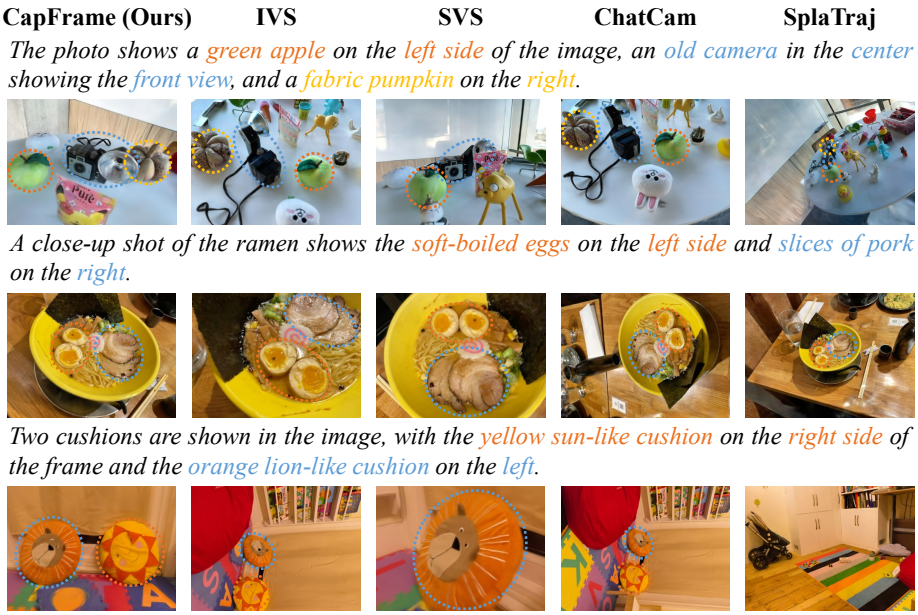

Fig. 4: Qualitative comparisons. CapFrame renders images aligned with specified subjects, composition and viewpoint, whereas other baselines are largely limited to object localization and often fail to satisfy composition and viewpoint requirements.

scale and select the frame that best matches the instruction.

Text-Instructed Viewpoint Grounding. As shown in Fig. 3, CapFrame grounds compositional text instructions to camera poses, enabling automatic viewpoint selection in diverse 3D Gaussian scenes. By leveraging the zero-shot reasoning ability of MLLMs, the framework generalizes to varied scenarios and supports intuitive text-guided exploration of complex 3D environments.

Comparisons with Baselines. Figure 4 presents qualitative comparisons under identical instructions. While IVS and SVS often retrieve viewpoints where the target subject is visible, they frequently fail to satisfy the required composition or orientation. For example, IVS places the *green apple* away from the left-side position, while SVS fails to place the *slices of pork* on the right. IVS can also inherit biases from retrieved camera poses: if the original views are tilted, interpolated viewpoints may preserve such unnatural orientations, as observed in the *two cushions* example. In the same example, ChatCam [33] produces a tilted view of the *cushions*, as CLIP-based [43] guidance is invariant to orientation. SplaTraj [34] settles on a viewpoint where the *cushions* are no longer visible, because its learned language field [42] fails to distinguish the target objects. In contrast, CapFrame optimizes camera poses using geometric pseudo labels encoding both layout and orientation, producing frames better aligned with the

Table 1: Quantitative comparisons. The left two columns report text-image similarity scores from two VLMs (CLIP and SigLIP2). The middle four columns report alignment scores (AS) and win rates (WR) from two MLLM judges (GPT and Gemini). The right two columns summarize user study results, including average rating and preference percentage. "—" indicates that the corresponding baselines are not evaluated in our user study.

Method	VLM metrics		MLLM judges				User study	
	CLIP↑	SigLIP2↑	GPT AS↑	GPT WR↑	Gemini AS↑	Gemini WR↑	Rating↑	Preference↑
SplaTraj [34]	0.241	0.276	3.08	3.0%	2.78	3.0%	—	—
ChatCam [33]	0.258	0.283	3.42	3.7%	3.27	5.2%	—	—
SVS	0.275	0.435	4.06	14.8%	3.88	16.3%	2.52	16.7%
IVS	0.265	0.396	3.49	5.9%	3.19	8.1%	2.53	5.6%
CapFrame (Ours)	**0.282**	**0.448**	**4.75**	**72.6%**	**4.22**	**67.4%**	**4.50**	**77.7%**

Table 2: SigLIP2 score before and after camera pose refinement.

Stage	Mip-NeRF	Deep Blending	Tanks	LERF-OVS	DL3DV-10K
Retrieve	0.121	0.274	0.307	0.503	0.393
Refine	**0.244**	**0.406**	**0.339**	**0.708**	**0.500**

compositional intent of the instruction. Quantitative results in Tab. 1 further show that CapFrame achieves the highest text-image alignment on VLM-based similarity. The same trend holds across two MLLM judges, where CapFrame achieves the best alignment score and win rate. Moreover, our user study reveals a consistent preference for CapFrame over IVS and SVS.

Component Analysis. We conduct studies to analyze the key components of CapFrame following the pipeline order: the Question-Evaluation (QE) strategy in **Retrieve**, the layout and orientation losses derived from geometric pseudo labels in **Translate**, and the pose refinement process in **Refine**.

We first evaluate QE in the **Retrieve** stage. As illustrated in Fig. 5, using only FG-CLIP [58] for coarse retrieval often returns semantically related views that do not contain the target subject, especially in cluttered scenes. Prompting the MLLM to assign a single holistic score shows similar limitations because compositional constraints are not explicitly evaluated. In contrast, QE decomposes the instruction into multiple questions, enabling the MLLM to assess candidate views along several semantic aspects. As a result, retrieved views more reliably contain the target subject and provide better initialization for pose optimization. Next, we conduct the ablation study to analyze the layout and orientation losses used in pose refinement (Fig. 6). With both losses, the optimized pose satisfies the desired composition and viewpoint. The layout loss regulates subject placement, while the orientation loss enforces the viewing direction. Removing either loss degrades alignment: without the orientation loss the camera fails to reach the specified view, and without the layout loss the subject placement deviates from

Table 3: Sensitivity analysis under different perturbations. We measure the deviation from the unperturbed output, where ΔR and Δt are the rotation and translation differences and ΔCLIP and ΔSigLIP2 are the changes in the VLM-based metrics.

Type	Perturbation	ΔR	Δt	ΔCLIP	ΔSigLIP2
Input text	Paraphrased text	9.61°	0.732	+0.013	+0.032
Top-K images	Top-3 images	8.99°	1.268	−0.014	−0.061
	Top-5 images	13.28°	1.801	−0.018	−0.165
Pseudo labels	Layout label removal	4.08°	5.701	−0.024	−0.236
	Orientation label removal	33.71°	2.312	−0.021	−0.128
	Moderate noise	8.39°	0.219	−0.012	−0.048

the intended composition. Finally, we examine the effect of **Refine** stage. Because the desired viewpoint may not exist among retrieved views, refinement performs gradient-based camera pose optimization using geometric pseudo labels, enabling continuous pose adjustment in $SE(3)$. We measure text-image alignment using SigLIP2 [51]. As shown in Tab. 2, refined views consistently achieve higher alignment scores than the retrieved initial views across all datasets.

Sensitivity Analysis. We analyze the sensitivity of CapFrame to different perturbations in Tab. 3, comparing the changes in camera pose and VLM-based metrics relative to the unperturbed output. First, paraphrasing each instruction twice with GPT-5.4-mini [39] slightly improves the alignment scores, demonstrating insensitivity to variation in input text. Next, when initializing from lower-ranked views (Top-3 and Top-5), the pose deviation grows gradually with K and the alignment scores decrease, indicating that camera refinement partially compensates for imperfect retrieval, though a good initialization remains beneficial. Finally, removing the pseudo labels reveals their complementary roles: excluding the layout label raises translation deviation, whereas excluding the orientation label increases rotation deviation. Moreover, under moderate label noise (5–10 pixels for layout and 5–10° for orientation), CapFrame stays close to its original output, confirming its robustness to imperfect pseudo labels.

7 Conclusion

We introduce a new task, text-instructed viewpoint grounding in 3D Gaussian scenes, and present CapFrame, a framework for solving it. CapFrame leverages the zero-shot reasoning of MLLMs to translate natural language instructions into geometric pseudo labels, including orientation and layout constraints, which guide camera pose refinement through differentiable optimization in 3DGS. Extensive experiments demonstrate that CapFrame generalizes across diverse scenes and produces viewpoints that more faithfully align with compositional text instructions, enabling intuitive language-driven exploration of 3D environments.

Fig. 5: Ablation of initialization. We present results of Top-K image selection across different retrieval methods. In complex scenes, (a) FG-CLIP [58] fails to localize the subject. Without QE for multi-dimensional scoring, the subject can still be absent from the image. (b) A good initialization is essential for camera pose convergence.

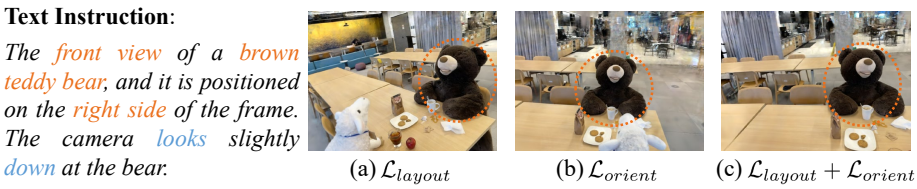

Fig. 6: Ablation of loss. Layout and orientation losses significantly influence the optimization performance. (a) Without orientation loss, the camera fails to achieve the front view of the bear. (b) Without layout loss, the bear is not positioned on the right side. (c) With both losses, the bear satisfies the specified composition and viewpoint.

Limitations. CapFrame relies heavily on MLLMs in several stages, including retrieval and translation. This design enables test-time optimization without additional training, but also introduces sensitivity to the prompts provided to the MLLM, which may affect the resulting pseudo labels and camera placement. While this work focuses on introducing the new task and establishing a first solution, future research could explore more robust prompting strategies and prompt optimization to further improve performance.

Acknowledgements

This work was supported by DENSO IT LAB Recognition, Control, and Learning Algorithm Collaborative Research Chair (Science Tokyo) and the experiments were conducted using TSUBAME 4.0 supercomputer.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
3. Blinn, J.: Where am i? what am i looking at? (cinematography). IEEE Computer Graphics and Applications **8**(4), 76–81 (2002)
4. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21476–21485 (2024)
5. Chen, Z., Wang, F., Wang, Y., Liu, H.: Text-to-3d using gaussian splatting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21401–21412 (2024)
6. Choi, S., Song, H., Kim, J., Kim, T., Do, H.: Click-gaussian: Interactive segmentation to any 3d gaussians. In: European Conference on Computer Vision. pp. 289–305 (2024)
7. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
8. Courant, R., Dufour, N., Wang, X., Christie, M., Kalogeiton, V.: E.t. the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In: European Conference on Computer Vision. pp. 464–480 (2024)
9. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. Advances in Neural Information Processing Systems **36**, 49250–49267 (2023)
10. Drucker, S.M., Galyean, T.A., Zeltzer, D.: Cinema: A system for procedural camera movements. In: Proceedings of the 1992 symposium on Interactive 3D graphics. pp. 67–70 (1992)
11. Galvane, Q., Christie, M., Lino, C., Ronfard, R.: Camera-on-rails: Automated computation of constrained camera paths. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games. pp. 151–157 (2015)
12. Galvane, Q., Ronfard, R., Christie, M., Szilas, N.: Narrative-driven camera control for cinematic replay of computer games. In: Proceedings of the 7th International Conference on Motion in Games. pp. 109–117 (2014)
13. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
14. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics **37**(6), 1–15 (2018)
15. Jain, U., Mirzaei, A., Gilitschenski, I.: Gaussiancut: Interactive segmentation via graph cut for 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 89184–89212 (2024)
16. Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H., Gao, F., Yang, Y., et al.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In: ACM SIGGRAPH. pp. 1–1 (2024)

17. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: European Conference on Computer Vision. pp. 371–386 (2018)
18. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3907–3916 (2018)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 139–1 (2023)
20. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: IEEE/CVF International Conference on Computer Vision. pp. 19729–19739 (2023)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
22. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)
23. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4), 1–13 (2017)
24. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International Conference on Machine Learning. pp. 19730–19742 (2023)
25. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning. pp. 12888–12900 (2022)
26. Li, L., Zhu, S., Fu, H., Tan, P., Tai, C.L.: End-to-end learning local multi-view descriptors for 3d point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1919–1928 (2020)
27. Li, W., Zhao, Y., Qin, M., Liu, Y., Cai, Y., Gan, C., Pfister, H.: Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. Advances in Neural Information Processing Systems **38**, 174306–174330 (2026)
28. Li, X., Lai, Z., Xu, L., Qu, Y., Cao, L., Zhang, S., Dai, B., Ji, R.: Director3d: Real-world camera trajectory and 3d scene generation from text. Advances in Neural Information Processing Systems **37**, 75125–75151 (2024)
29. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
30. Lino, C., Christie, M.: Intuitive and efficient camera control with the toric space. ACM Transactions on Graphics **34**(4), 1–12 (2015)
31. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (2024), `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, accessed: 2026-06-26
32. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision. pp. 38–55 (2024)
33. Liu, X., Tai, Y.W., Tang, C.K.: Chatcam: Empowering camera control through conversational ai. Advances in Neural Information Processing Systems **37**, 54483–54506 (2024)

34. Liu, X., Zhang, T., Johnson-Roberson, M., Zhi, W.: Splatraj: Camera trajectory generation with semantic gaussian splatting. arXiv preprint arXiv:2410.06014 (2024)
35. Liu, Z., Wang, Y., Zheng, S., Pan, T., Liang, L., Fu, Y., Xue, X.: Reasongrounder: Lvlm-guided hierarchical feature splatting for open-vocabulary 3d visual grounding and reasoning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3718–3727 (2025)
36. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: International Conference on 3D Vision. pp. 800–809 (2024)
37. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18039–18048 (2024)
38. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
39. OpenAI: Introducing GPT-5.4 mini and nano (2026), `https://openai.com/index/introducing-gpt-5-4-mini-and-nano/`, accessed: 2026-06-26
40. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
41. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems **30** (2017)
42. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20051–20060 (2024)
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
44. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)
45. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
46. Schwarz, K., Sauer, A., Niemeyer, M., Liao, Y., Geiger, A.: Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. Advances in Neural Information Processing Systems **35**, 33999–34011 (2022)
47. Shi, J.C., Wang, M., Duan, H.B., Guan, S.H.: Language embedded 3d gaussians for open-vocabulary scene understanding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5333–5343 (2024)
48. Shi, J., Jiang, X., Guillemot, C.: Learning fused pixel and feature-based view reconstructions for light fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2555–2564 (2020)
49. Skartados, E., Yucel, M.K., Manganelli, B., Drosou, A., Saà-Garriga, A.: Finding waldo: Towards efficient exploration of nerf scene spaces. In: Proceedings of the 15th ACM Multimedia Systems Conference. pp. 155–165 (2024)
50. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022)

51. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding. Localization, and Dense Features **6** (2025)
52. Wang, X., Courant, R., Shi, J., Marchand, E., Christie, M.: Jaws: Just a wild shot for cinematic transfer in neural radiance fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16933–16942 (2023)
53. Wang, Z., Zhang, Z., Pang, T., Du, C., Zhao, H., Zhao, Z.: Orient anything: Learning robust object orientation estimation from rendering 3d models. International Conference on Machine Learning (2025)
54. Wen, M., Wu, S., Wang, K., Liang, D.: Intergsedit: Interactive 3d gaussian splatting editing with 3d geometry-consistent attention prior. In: IEEE/CVF International Conference on Computer Vision. pp. 26136–26145 (2025)
55. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20310–20320 (2024)
56. Wu, J., Bian, J.W., Li, X., Wang, G., Reid, I., Torr, P., Prisacariu, V.A.: Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In: European Conference on Computer Vision. pp. 55–71 (2024)
57. Wu, Y., Meng, J., Li, H., Wu, C., Shi, Y., Cheng, X., Zhao, C., Feng, H., Ding, E., Wang, J., et al.: Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. Advances in Neural Information Processing Systems **37**, 19114–19138 (2024)
58. Xie, C., Wang, B., Kong, F., Li, J., Liang, D., Zhang, G., Leng, D., Yin, Y.: Fg-clip: Fine-grained visual and textual alignment. In: International Conference on Machine Learning (2025)
59. Yang, D., Wang, X., Gao, Y., Liu, S., Ren, B., Yue, Y., Yang, Y.: Opengs-fusion: Open-vocabulary dense mapping with hybrid 3d gaussian splatting for refined object-level understanding. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 21135–21142 (2025)
60. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The dawn of lmms: Preliminary explorations with gpt-4v (ision). arXiv preprint arXiv:2309.17421 (2023)
61. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. In: European Conference on Computer Vision. pp. 162–179 (2024)
62. Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6796–6807 (2024)
63. Zhai, H., Zhang, X., Zhao, B., Li, H., He, Y., Cui, Z., Bao, H., Zhang, G.: Splatloc: 3d gaussian splatting-based visual localization for augmented reality. IEEE Transactions on Visualization and Computer Graphics (2025)
64. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023)
65. Zhang, M., Wu, T., Tan, J., Liu, Z., Wetzstein, G., Lin, D.: Gendop: Auto-regressive camera trajectory generation as a director of photography. In: IEEE/CVF International Conference on Computer Vision. pp. 18229–18239 (2025)

66. Zhao, H., Wang, H., Zhao, X., Fei, H., Wang, H., Long, C., Zou, H.: Physsplat: Efficient physics simulation for 3d scenes via mllm-guided gaussian splatting. In: IEEE/CVF International Conference on Computer Vision. pp. 5242–5252 (2025)
67. Zhou, S., Fan, Z., Xu, D., Chang, H., Chari, P., Bharadwaj, T., You, S., Wang, Z., Kadambi, A.: Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In: European Conference on Computer Vision. pp. 324–342 (2024)