RelationVGGT: Visual Geometry Transformers for 3D Spatial Relation Segmentation

Minsu Kim[1] **Jaesung Choe**[2] **Jiwoo Lee**[1]
Yu-Chiang Frank Wang[2] **Seon Joo Kim**[1]

[1]**Yonsei University** [2]**NVIDIA**

Project Page

Abstract

Recent advances in 3D reconstruction have progressed from per-scene optimization to feed-forward inference, and semantic scene understanding has followed suit—yet existing methods remain confined to object-centric perception, neglecting spatial relations between objects. We formulate *3D spatial relation segmentation* in a feed-forward, pose-free multi-view setting: given a visually specified subject and a relational text query, the model segments the target across views without receiving its category name. To this end, we propose RelationVGGT, a novel feed-forward framework that integrates semantic features from a visual foundation model with geometry-aware representations from a 3D geometry foundation model and leverages a relation transformer for subject-conditioned, cross-view relation prediction—requiring neither per-scene optimization nor known camera poses. We additionally provide a fully automated annotation pipeline built on ScanNet++ with VLMs and LLMs, enabling scalable training data generation for this new task.

1 Introduction

Recent years have witnessed remarkable advancements in 3D reconstruction and neural rendering. Pioneering representations such as Neural Radiance Fields (NeRF) [1] and 3D Gaussian Splatting (3DGS) [2] have revolutionized scene-optimized view synthesis and geometric modeling. More recently, visual geometry models such as DUSt3R [3] and VGGT [4] have demonstrated that dense 3D geometry can be predicted directly from multi-view images without camera pose priors or Structure-from-Motion pipelines, fundamentally shifting how multi-view geometry can be inferred.

This shift has not been limited to reconstruction: semantic scene understanding has followed the same trajectory. Methods such as LangSplat [5] and OpenGS [6] embed vision-language features into scene-optimized 3D primitives, enabling open-vocabulary 3D scene understanding and zero-shot object segmentation from simple textual queries like ‘chair’ or ‘table’. Beyond per-scene methods, feed-forward approaches like PanSt3R [7] and Uni3R [8] integrate geometry foundation model [3, 4, 9] priors to achieve multi-view consistent semantic segmentation, further bypassing the need for per-scene optimization and posed images.

While these previous studies have made significant strides, they dominantly focus on object-centric scene understanding—essentially treating environments as mere collections of isolated entities. However, holistic scene perception requires a deeper understanding of how objects interact. We understand an environment by recognizing how objects are spatially arranged and related to one another—a monitor *resting on* a desk, a mug *next to* a keyboard. Such relational reasoning underpins higher-order tasks such as spatial question answering and is a core capability that embodied agents must acquire to act in 3D environments. This motivates us to study relational 3D reasoning beyond object-level perception, particularly in a generalizable multi-view setting.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To this end, we formulate the task of *3D spatial relation segmentation* in a feed-forward, pose-free setting, where the goal is to localize, across scene views, the object satisfying a spatial relationship determined from the underlying 3D structure. Given multi-view images, a subject mask in a reference view, and a relational text query (e.g, "resting on"), the model must predict segmentation masks of the target object consistently across all target views. This formulation goes beyond conventional 3D segmentation by requiring semantic identity, directional spatial relation reasoning, and cross-view geometric consistency.

Existing approaches offer two natural starting points for this formulation. RelationField [10] learns inter-object relationships on top of radiance fields, but relies on scene-specific optimization with calibrated camera poses, limiting its scalability and generalizability to unseen scenes. Alternatively, video-language models perform multi-view relational reasoning, but without a 3D scene representation, they struggle to maintain object correspondences under large viewpoint changes.

Motivated by these limitations, we propose **RelationVGGT**, a feed-forward framework for 3D spatial relation segmentation that requires neither per-scene optimization nor known camera poses. Our core idea is to reformulate the volumetric relation search of prior work as a per-pixel relation prediction problem on the target-view image grid, conditioned on the reference subject. To this end, we fuse semantic features from DINOv2 [11] with geometry-aware representations from the Pi3 [9] encoder, and employ a relation transformer that alternates subject-conditioned cross-attention with multi-view self-attention to produce language-aligned relation features for open-vocabulary segmentation. A critical challenge is the scarcity of training data with multi-view consistent relation annotations. To address this, we build a fully automated data generation pipeline on top of ScanNet++ [12], exploiting instance-level labels and VLM/LLM-based relation extraction to produce scalable training data.

In summary, our contributions are threefold. First, we formulate *3D spatial relation segmentation* in a feed-forward, pose-free multi-view setting, requiring models to localize target objects defined by spatial relations consistently across views. Second, we propose RelationVGGT, a feed-forward architecture that fuses 2D semantic and 3D geometry-aware features and reformulates volumetric relation search as per-pixel, subject-conditioned relation prediction. Third, we provide a fully automated relation annotation pipeline leveraging VLMs and LLMs on top of ScanNet++, enabling scalable training data generation.

2 Related Works

3D Scene Understanding with Foundation Models. Recent advances in 3D scene understanding span scene-optimized representations such as NeRF [1] and 3DGS [2], as well as feed-forward geometry foundation models [3, 4, 9, 13, 14, 15] that predict dense 3D geometry directly from multi-view images without known camera poses. Building on these, methods such as LangSplat [5] and follow-up works [6, 16, 17, 18, 19] embed vision-language features into 3D primitives for open-vocabulary segmentation, while feed-forward approaches [7, 8] achieve multi-view consistent semantics without per-scene optimization. However, these methods remain object-centric, localizing individual entities without reasoning about inter-object spatial relations. Our work builds on these foundations, fusing DINOv2 [11] semantics with geometry-aware representations to enable relational reasoning beyond object-level perception.

3D Spatial Relation and Scene Graph. Understanding relationships between objects has traditionally been formulated as Scene Graph Generation (SGG), where nodes represent objects and edges encode pairwise predicates. While 2D SGG methods [20, 21, 22] have been widely studied, they cannot capture the multi-view nature of 3D environments. 3D scene graph methods [23, 24, 25] construct graphs from 3D point clouds and jointly predict object nodes and relation edges. However, these approaches assume access to pre-segmented object instances and focus on classifying relations from a fixed predicate vocabulary. More recently, RelationField [10] has pioneered relational understanding in neural radiance fields by learning continuous relation features conditioned on pairs of 3D coordinates. However, it requires scene-specific optimization with calibrated camera poses and localizes relations through queries over the reconstructed 3D representation. In contrast, we replace 3D relation search with feed-forward per-pixel prediction over unposed multi-view images. More broadly, we take a relation-first perspective: rather than recovering object nodes and then classifying their edges, we directly segment the target conditioned on a visually specified subject and a relation

query. This formulation is complementary to scene graph generation and could be integrated with node prediction methods to jointly recover full scene graphs in future work.

Visual Grounding. VLM-based methods such as LISA [26] enable open-vocabulary visual grounding from natural-language queries, while Video-LISA [27] extends this capability to video inputs. Recent approaches such as SA2VA [28] and UniPixel [29] further support dense pixel-level mask prediction for grounded targets. Despite their strong grounding capability, these methods do not explicitly model the underlying 3D scene structure and are primarily designed for image or temporally continuous video observations, making relational grounding more challenging under sparse, wide-baseline multi-view inputs.

Closely related 3D referring segmentation methods [30, 31, 32, 33] localize targets from language descriptions in 3D or multi-view representations. Their formulation, however, differs from ours: a textual referring expression identifies the target object, potentially through its category, attributes, or relation to other objects. In contrast, we visually specify the reference subject with a mask and provide only the relation as text; the target identity and category are not given and must be inferred from the subject–relation pair.

3 Method

3.1 3D Spatial Relation Segmentation

Problem Definition and Notations. We introduce *3D spatial relation segmentation*, a task that aims to localize, across multiple views of a scene, the object satisfying a spatial relationship that can only be determined from the underlying 3D structure of the scene. Here, spatial relations refer to relationships arising from the 3D arrangement of objects, such as support, containment, and proximity. Consider a scene captured by a set of N images $\mathcal{I} = \{I_i\}_{i=1}^{N}$. One image I_r is designated as the *reference view*, and the remaining images are treated as *target views*, denoted by $\mathcal{I}_{\mathrm{tgt}} = \mathcal{I} \setminus \{I_r\}$. In the reference view, a binary mask M_s indicates the *subject*, i.e., the query object from which the relation originates. A relational text query q specifies the spatial relationship that the desired object should satisfy with respect to the subject. We refer to this desired object as the *target object*. For example, if the subject mask corresponds to a monitor and the relation query is "resting on," the target object is the desk supporting the monitor. The goal is to predict, for each target view $I_t \in \mathcal{I}_{\mathrm{tgt}}$, a segmentation mask that localizes the target object from the viewpoint of I_t.

Motivation. A natural baseline for this task is RelationField [10], which extends a radiance field with an additional query coordinate to model relational semantics:

$$g_\theta(\mathbf{x}, \mathbf{z}) \;\mapsto\; \mathbf{r}, \tag{1}$$

where $\mathbf{x}$ is the target 3D position, $\mathbf{z}$ the query 3D position, and $\mathbf{r}$ the predicted relation feature. The two positions are encoded via positional encodings and concatenated as input to the MLP network. This design has two fundamental limitations. First, because both the query and target are represented as 3D coordinates, locating the target object requires evaluating g_θ at every candidate position in the scene volume. For example, if the subject is a monitor and the relation is "resting on," the model should identify the desk supporting it—yet to do so, it must sweep through the entire 3D space, a search that is both computationally expensive and inherently dependent on an explicit volumetric reconstruction. Second, the reliance on scene-specific positional encodings ties the learned representations to a particular coordinate system, leaving generalization to unseen scenes unexplored.

Our key observation is that this volumetric search is unnecessary. Given the subject mask M_s in the reference view, we can formulate target localization on the image grid of each target view, rather than over a dense 3D volume. This replaces an unstructured volumetric search with a per-pixel relation prediction problem conditioned on the reference subject, without relying on a scene-specific 3D field at test time. Building on this observation, we define a per-view *relation feature map*:

$$\mathcal{R}_t = \big\{\, \mathbf{r}_t(\mathbf{u}) \;\big|\; \mathbf{u} \in I_t \big\}, \tag{2}$$

where $\mathbf{r}_t(\mathbf{u}) \in \mathbb{R}^D$ encodes the spatial relationship between location $\mathbf{u}$ in target view I_t and the reference subject specified by M_s. By aligning this feature with the language embedding space, open-vocabulary relational queries can be evaluated through feature–text similarity, yielding dense relevance maps from which segmentation masks are derived.

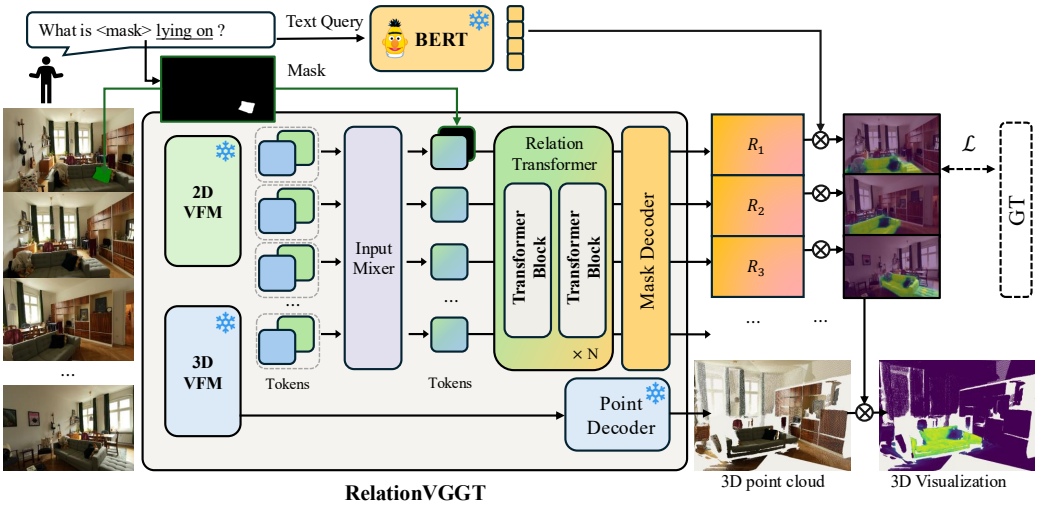

Figure 1: **Overview of RelationVGGT.** Multi-view images are encoded by frozen 2D and 3D visual foundation models and fused via an input mixer. The subject mask conditions the relation transformer, whose output is decoded into per-view relation feature maps. The target object is segmented via feature–text similarity with the BERT-encoded relational query.

Formally, our model learns a function f_θ:

$$f_\theta(I_r,\, M_s,\, \mathcal{I}_{\text{tgt}},\, \mathbf{u}) \;\to\; \mathcal{R}_t, \tag{3}$$

which predicts the relation embedding at each target-view location conditioned on the reference subject. Given the query embedding $\mathbf{e}(q)$, the target object is segmented by measuring compatibility between $\mathbf{r}_t(\mathbf{u})$ and $\mathbf{e}(q)$ over all $\mathbf{u} \in \Omega_t$. This reformulation eliminates the need for volumetric search and scene-specific 3D optimization, while supporting generalization to novel scenes through multi-view-aware representations.

3.2 Architecture

The overall pipeline of RelationVGGT is illustrated in Fig. 1. Given a reference image I_r with a subject mask M_s and target images $\mathcal{I}_{\text{tgt}}$, our model extracts dense patch-level representations for all views, performs subject-conditioned relation reasoning across views, and decodes pixel-aligned relation features for open-vocabulary segmentation.

Feature extraction. Relational reasoning requires both object-level semantics and spatially grounded scene structure. Following [7], we extract two complementary dense representations from the input image set $\mathcal{I}$. For semantic cues, we use DINOv2 [11] feature maps, which provide dense patch-level visual representations with strong object-level semantics. Geometry-aware cues are obtained from the Pi3 [9] encoder tokens before its task-specific prediction heads, preserving general multi-view geometric information. The two feature streams are represented on a common patch grid and subsequently fused by the input mixer.

For each image $I_n \in \mathcal{I}$, the two feature streams are denoted as

$$E_n^D = \text{Enc}_D(I_n), \qquad E_n^P = \text{Enc}_P(\mathcal{I})_n. \tag{4}$$

We concatenate them along the channel dimension and map them to a compact joint representation using the input-mixing architecture of [7]:

$$F_n = M_\theta\big([E_n^D;\, E_n^P]\big) \in \mathbb{R}^{P\times d_f}, \tag{5}$$

where M_θ is the input mixer, $[\cdot\,;\,\cdot]$ denotes channel-wise concatenation, and $P = H_p W_p$ is the number of patches per image. The mixer consists of a linear projection followed by a small stack of transformer blocks with RoPE [34] positional encoding. This produces joint semantic-geometric tokens F_n for each view, which are passed to the relation reasoning stage.

Relation prediction. To predict a per-pixel relation feature map for each target view, the model must condition target-view representations on the reference subject. A key design choice is how to represent the subject. Rather than compressing it into a single pooled vector, we retain the full set of reference-view patch tokens and mask out non-subject patches, thereby preserving the subject's spatial layout and fine-grained semantics. Concretely, let $F_r \in \mathbb{R}^{P \times d_f}$ denote the reference-view tokens and $\bar{M}_s \in \{0,1\}^P$ be the subject mask downsampled to the patch grid. The subject-masked reference tokens are formed as

$$F_r^s = \bar{M}_s \odot F_r \in \mathbb{R}^{P \times d_f}, \tag{6}$$

where $\odot$ denotes element-wise multiplication, zeroing out non-subject patches. Target-view tokens are stacked into a single sequence for joint processing:

$$Z^0 = \mathrm{Concat}\big(\{F_t\}_{I_t \in \mathcal{I}_{\mathrm{tgt}}}\big) \in \mathbb{R}^{(N-1)P \times d_f}. \tag{7}$$

The relation transformer then refines this sequence by alternating subject-conditioned cross-attention and global self-attention:

$$\bar{Z}^\ell = \mathrm{CrossAttn}_\ell\big(Z^\ell,\, F_r^s\big), \qquad Z^{\ell+1} = \mathrm{SelfAttn}_\ell\big(\bar{Z}^\ell\big). \tag{8}$$

Cross-attention allows each target patch to attend to the subject-masked reference tokens, while self-attention enables information exchange among target patches across all views. This makes the resulting representation both subject-conditioned and multi-view-aware.

After L layers of the relation transformer, the updated tokens are reshaped back into per-view patch grids. Since the transformer operates at patch resolution, a mask decoder upsamples the tokens into a pixel-aligned relation feature map:

$$\mathcal{R}_t = \mathrm{Dec}\big(Z_t^L\big) = \big\{\, \mathbf{r}_t(\mathbf{u}) \;\big|\; \mathbf{u} \in I_t \big\}, \tag{9}$$

where Z_t^L denotes the final tokens corresponding to target view I_t, and $\mathcal{R}_t$ is the resulting relation feature map aligned with the language embedding space. In practice, we implement the mask decoder using a DPT head [35]. The target object is then localized by computing feature–text similarity with the query embedding $\mathbf{e}(q)$, as detailed in Sec. 3.4.

3D geometry. Additionally, we use the Pi3 [9] point decoder to predict a point map and confidence map for each target view. Since $\mathcal{R}_t$ is pixel-aligned, each relation feature $\mathbf{r}_t(\mathbf{u})$ can be associated with its corresponding 3D point, yielding a confidence-aware 3D relation representation across views.

3.3 Training

The spatial relationship between two objects can often be described by multiple valid expressions. For example, the relational query "standing on" may equally well be expressed as "on" or "supported by." Treating this as a single-label classification problem may penalize semantically plausible alternatives. To address this, we supervise the same target mask with multiple compatible relational queries, treating each query independently with a sigmoid focal loss [36]. The full training objective combines this multi-query focal loss with a Dice loss for spatial mask alignment:

$$\mathcal{L}_{\mathrm{mask}} = w_{\mathrm{focal}}\, \mathcal{L}_{\mathrm{focal}}^{\text{multi-q}} + w_{\mathrm{dice}}\, \mathcal{L}_{\mathrm{dice}}. \tag{10}$$

We provide further details of the multi-query supervision and training objective in the appendix.

3.4 Inference

Given the predicted relation feature map $\mathcal{R}_t$, we compute a relevance logit $z_t(\mathbf{u}; q)$ at each pixel by matching the relation feature with the text query embedding:

$$z_t(\mathbf{u}; q) = \mathbf{r}_t(\mathbf{u})^\top \mathbf{e}(q). \tag{11}$$

The corresponding mask score is obtained by applying sigmoid to the logits:

$$S_t(\mathbf{u}; q) = \mathrm{sigmoid}\left(z_t(\mathbf{u}; q)\right). \tag{12}$$

Unlike feature field methods that require negative-prompt normalization, we empirically find that directly applying sigmoid suffices without softmax over positive and negative text embeddings. The final binary mask is obtained using a fixed threshold $\tau = 0.3$:

$$\hat{M}_t(q) = \{\mathbf{u} \,|\, S_t(\mathbf{u}; q) \geq \tau\}. \tag{13}$$

We use the same threshold for all experiments without additional post-processing.

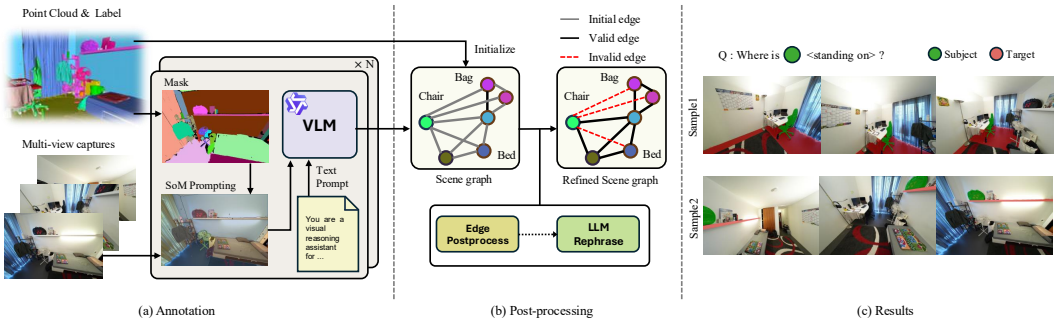

Figure 2: **Overview of the relation annotation pipeline.** (a) Instance masks are projected onto frames with SoM prompting to query a VLM for pairwise relations. (b) Proposals are aggregated into a scene graph and refined via geometry-aware postprocessing and LLM. (c) Example results showing multi-view consistent subject (green) and target (red) annotations.

3.5 Dataset

Our model requires multi-view consistent relation labels with object-level masks, but such supervision is scarce and costly to annotate manually. We therefore build an automatic annotation pipeline on ScanNet++ [12], which provides diverse indoor scenes with 3D instance-level point annotations. The overview of the pipeline is shown in Fig. 2.

For each scene, we initialize a scene graph with object instances as nodes and edges representing pairwise spatial relations, initially left empty. We project instance masks onto posed RGB frames, apply Set-of-Mark (SoM) prompting to mark visible objects, and query a VLM to obtain frame-wise relation proposals. These proposals are collected as candidate edges but are not yet treated as valid relations, as they must pass the following multi-view and geometric verification steps.

Since frame-wise VLM proposals are often noisy, we filter unreliable predictions by requiring agreement across multiple viewpoints. For each object pair visible across multiple frames, the VLM independently proposes relations from each viewpoint. We count how many frames predict each relation for a given pair. If the count falls below $K{=}3$, the relation is rejected as insufficiently supported. Otherwise, it is retained as a candidate edge. Since we already know which frames each subject-target pair appears in via rasterization of 3D instance points, retained relations are propagated to all co-visible frames, completing the multi-view annotation.

We further apply a geometry-based filtering step to remove geometrically wrong predictions. Using 3D bounding boxes and instance centroids, we reject relations between object pairs whose bounding boxes do not overlap and whose centroids are farther than a distance threshold, such as contact or support relations predicted between distant objects.

We observe that the VLM tends to be conservative, repeatedly producing a narrow set of canonical relation descriptions to maintain accuracy. To compensate for this limited lexical variety, an LLM rephrases each validated triplet (subject, relation, target) into diverse linguistic forms, aiming to preserve the subject-target correspondence while increasing query diversity. For example, the triplet (TV, hanging on, wall) can be rephrased into queries such as “mounted on” or “installed on.”

4 Experiments

Implementation details. For the automated annotation pipeline described in Sec. 3.5, we use Qwen3-VL-8B [37] as the VLM and GPT-5.2 for predicate rephrasing. For text-query encoding, we use Jina Embeddings v3, following RelationField [10]. The relation transformer follows the ViT-S configuration [38], including the transformer depth, number of attention heads, and hidden dimensionality. Our model outputs dense relation predictions at a spatial resolution of 512×388. We train our model on 900 ScanNet++ [12] scenes annotated by the proposed dataset pipeline. Training is performed for 200 epochs on 4 NVIDIA H200 GPUs with a per-GPU batch size of 7. We use AdamW with a learning rate of 1×10^{-4} and cosine learning-rate decay.

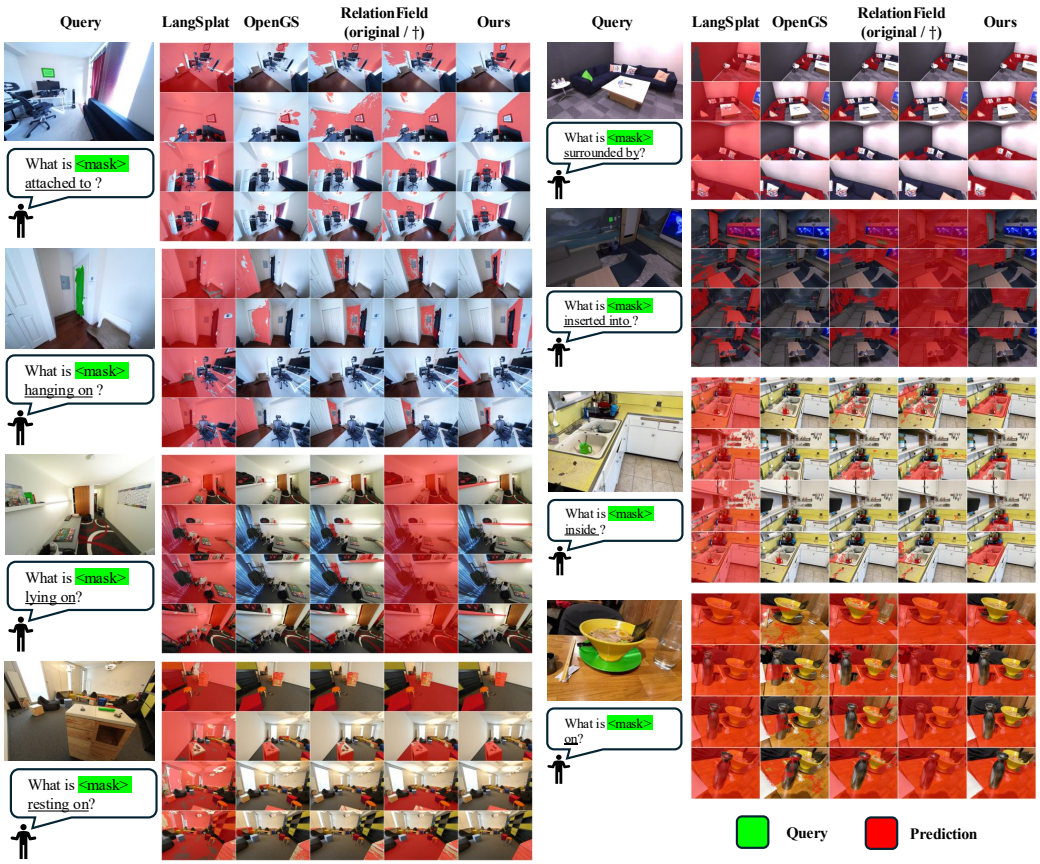

Figure 3: Qualitative results on the ScanNet++, Replica, LERF benchmarks. Please zoom for the detail. Predicted masks are overlaid in red on the target views.

Baselines. We compare against representative feature-field methods for open-vocabulary 3D scene understanding. LangSplat [5] and OpenGaussians [6] are CLIP-based methods for open-vocabulary object segmentation; at inference time, we directly use the full relation query as the text prompt, since they do not explicitly model object-predicate structure. RelationField is the closest baseline, as it supports both object-level and predicate-level querying. It first localizes the target object using the object field and then refines the prediction using the relation embedding. For a more isolated evaluation of relation modeling, we also evaluate RelationField† [10], which removes the target-object retrieval stage and localizes regions using only the relation embedding. Notably, all feature-field baselines require posed RGB images and per-scene optimization, whereas our feed-forward model directly predicts relation fields from unposed multi-view images.

Benchmarks. We build a manually annotated benchmark on ScanNet++ [12], LERF [39], and Replica [40] to evaluate relational understanding in 3D scenes. For ScanNet++, we use the 26 samples proposed by RelationField and add 14 newly annotated samples. For LERF and Replica, we annotate 31 relations from 3 scenes and 48 relations from 8 scenes, respectively. Frames are sampled with a moderate stride to avoid overly similar views and ensure sufficient viewpoint diversity. The resulting benchmark covers 28 unique predicates. We report mIoU as the primary metric and additionally report mAcc, which measures the fraction of queries whose IoU exceeds a threshold of 0.25, providing a complementary view of per-query success rate.

4.1 Quantitative results

Table. 1 summarizes the segmentation performance across three benchmark splits. Our method achieves the highest average mIoU of 0.382 and mAcc of 0.549, outperforming the strongest baseline, RelationField, by substantial margins in both mIoU and mAcc. Notably, our method does not rely on target object queries at test time and instead leverages relational reasoning alone, which is a key distinction from prior approaches.

Method	Replica		LERF		ScanNet++		Avg.	
	mIoU (↑)	mAcc (↑)	mIoU (↑)	mAcc (↑)	mIoU (↑)	mAcc (↑)	mIoU (↑)	mAcc (↑)
LangSplat [5]	0.191	0.256	0.159	0.254	0.074	0.064	0.141	0.191
OpenGaussian [6]	0.349	0.470	0.256	0.378	0.162	0.254	0.255	0.367
RelationField [10]	0.326	0.537	0.276	0.410	0.307	0.491	0.303	0.480
RelationField†	0.219	0.336	0.232	0.346	0.240	0.395	0.230	0.359
Ours	0.446	0.640	0.254	0.332	0.445	0.675	0.382	0.549

Table 1: Quantitative comparison on the Replica, LERF, and ScanNet++ benchmarks. RelationField† denotes the relation-only variant without target-object retrieval; Avg. is the macro average across the three benchmarks.

Method	**Small** (40)		**Medium** (39)		**Large** (40)	
	mIoU (↑)	mAcc (↑)	mIoU (↑)	mAcc (↑)	mIoU (↑)	mAcc (↑)
VideoLISA [27]	0.137	0.171	0.105	0.143	0.038	0.064
SA2VA [28]	0.267	0.321	0.225	0.289	0.192	0.225
UniPixel [29]	0.423	0.496	0.328	0.451	0.354	0.475
Ours	0.417	0.550	0.404	0.640	0.366	0.575

Table 2: Comparison with video foundation models across camera-baseline ranges. Small, Medium, and Large denote near-equal groups of the 119 queries ordered by camera baseline. All methods are given the subject region and relation query without the target category; video baselines process the multi-view images as a video sequence.

On the Replica-based and ScanNet++-based splits, where camera trajectories broadly cover entire indoor scenes rather than focusing on individual objects, open-vocabulary segmentation becomes particularly challenging. Our method consistently outperforms all baselines on both splits, demonstrating its robustness in handling diverse viewpoints and cluttered environments.

On the LERF-based split, all methods exhibit lower overall performance due to the densely cluttered tabletop scenes, where numerous small objects are tightly co-located and relational reasoning among them becomes inherently more difficult. Even under this challenging setting, our method remains comparable to RelationField, which has access to ground-truth category names at test time. Furthermore, our method outperforms RelationField†, which relies solely on relation embeddings without object retrieval, confirming that our relational modeling captures meaningful inter-object structure even in cluttered scenarios.

Comparison with video foundation models. As shown in Table. 2, we additionally compare against video foundation models by providing the subject region and relation query as input (omitting the target category name) and treating multi-view frames as video input. We sort all benchmark samples by camera baseline and divide them into three equal-sized groups. Among the compared methods, UniPixel achieves competitive mIoU on the small-baseline setting, but performance generally degrades as the baseline widens, since larger viewpoint changes break the temporal continuity assumptions underlying video representations. In contrast, our method maintains stable performance across all baseline ranges. We attribute this to the 3D foundation model backbone, which provides multi-view-aware features encoding scene structure rather than relying on frame-to-frame correspondence. The result is visualized in Fig. 4.

4.2 Qualitative results

Fig. 3 presents qualitative comparisons against baselines. LangSplat and OpenGaussian produce fragmented masks with significant spillover into background regions, as CLIP-based features alone lack the granularity to localize relation-relevant areas. RelationField yields more focused predictions but still exhibits noisy boundaries, while RelationField† tends to activate broadly across the scene due to the absence of target-object retrieval. In contrast, our method produces notably cleaner mask boundaries that tightly conform to the target region.

Moreover, despite not employing any explicit 3D representation such as radiance fields or 3D Gaussian splatting, our predictions remain visually consistent across viewpoints with substantial

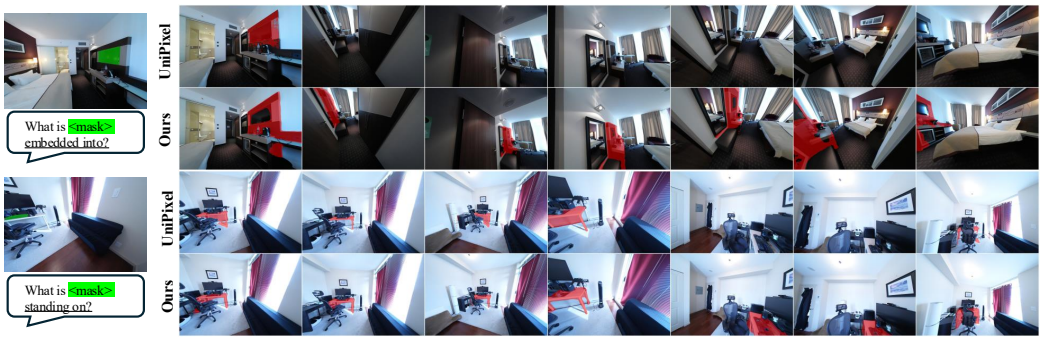

Figure 4: Qualitative comparison between our method and UniPixel. Our method produces view-consistent masks, whereas UniPixel fails to maintain consistency across views. Please zoom for the detail. Predicted masks are overlaid in red on the target views.

changes in camera pose. We attribute this to the 3D foundation model backbone, which enables the relation transformer to implicitly capture view-invariant relational semantics from multi-view input without explicit 3D reconstruction.

4.3 Ablations

Variants of input features. As an alternative source of geometric cues, we consider using 3D pointmaps, which explicitly represent scene geometry as per-pixel 3D coordinates. Following [41], we apply standardization to account for scene scale variations, and then use positional encoding to project the coordinates into a higher-dimensional space. We observe that employing 3D pointmaps leads to a performance drop. We hypothesize that this is because raw pointmap coordinates, while geometrically informative, lack the semantic cues already embedded in learned features. Removing DINOv2 [11] features leads to a notable drop, as the model loses access to the rich object-level semantics that DINOv2 provides. The degradation is even more pronounced without Pi3 features, since DINOv2 alone lacks multi-view awareness and cannot capture cross-view geometric correspondence.

Variants of relation decoder architecture. To validate the design of the relation transformer, we compare against two alternative attention block configurations: a cross-attention-only variant and the alternating attention block proposed by [4], which interleaves self-attention and per-frame attention instead of our design. As shown in Table 3, both variants lead to lower performance, demonstrating the effectiveness of our attention block design.

Training objectives. Replacing our multi-query sigmoid focal loss with a single-label sigmoid focal loss leads to a notable performance drop, confirming that spatial relations are inherently many-to-one. By allowing multiple valid predicates to supervise each target mask simultaneously, the model learns a more robust relational representation and avoids being penalized for semantically plausible alternatives.

Design	mIoU↑	mAcc↑
Ours	**0.445**	**0.675**
Input feature variants		
P.E(pointmap) + DINOv2	0.370	0.568
pi3 only	0.422	0.579
DINOv2 only	0.370	0.579
Architecture variants		
cross-attn only	0.399	0.579
alternating-attention [4]	0.394	0.571
Training objectives		
single-label sigmoid focal loss	0.406	0.604
Annotation pipeline		
w/o geometric edge filtering	0.398	0.625

Table 3: Ablation study on model design and the annotation pipeline, evaluated on ScanNet++ benchmark (40 queries) with similarity threshold $\tau = 0.3$.

Effect of dataset pipeline. As shown in Table 3, removing geometry-based edge filtering causes a performance drop, as the model receives physically implausible supervision such as contact relations between distant objects. Additional dataset statistics and analysis are provided in the appendix.

Subject mask	Mask IoU ↑	mIoU ↑	mAcc ↑
GT mask	1.000	0.382	0.549
SAM2, one click	0.871	**0.383**	**0.551**
SAM2, one box	0.923	0.382	0.549
Qwen3-VL + SAM2	0.719	0.357	0.512
GroundingDINO + SAM2	0.587	0.322	0.454

Table 4: Robustness to subject-mask specification across three benchmarks (119 queries). All variants use the same RelationVGGT checkpoint without retraining.

Predicate expression	n	ΔmIoU ↑	ΔmAcc@0.25 ↑
Meaning-preserving synonyms (set A)	119	-0.006	-0.006
Meaning-preserving synonyms (set B)	119	-0.004	-0.001
Rare synonym expressions	113	-0.022	-0.042
Unseen synonym expressions	119	-0.031	-0.040

Table 5: Robustness to predicate expressions. We report paired per-query changes relative to the original predicate queries. The two synonym sets use independently constructed meaning-preserving alternatives. Rare expressions use the least frequent suitable synonym observed in training (stem frequency ≤ 20 for 94/113 queries; six queries without a suitable alternative are excluded), whereas unseen expressions use synonyms whose stems never occur during training.

Robustness to subject-mask specification. We replace the ground-truth subject mask with promptable or automatically estimated masks without retraining the model. As shown in Table 4, single-click or box prompting with SAM2 preserves essentially the same performance as the ground-truth mask despite imperfect mask overlap, indicating that precise subject boundaries are not required when the intended instance is correctly specified. We further evaluate fully automatic subject grounding using Qwen3-VL-8B+SAM2 and GroundingDINO+SAM2. The former retains most of the ground-truth-mask performance without manual annotation, whereas the latter shows a larger degradation. We attribute this primarily to subject-instance ambiguity rather than boundary precision, as 58% of reference views contain multiple instances of the queried category.

Robustness to predicate expressions. We evaluate lexical robustness by replacing the benchmark predicates with meaning-preserving alternatives while keeping the scene, subject mask, and target fixed. As shown in Table 5, the model remains robust to paraphrased predicate expressions, with only minor degradation for common synonyms and limited degradation even for rare or previously unseen expressions. These results indicate that relation prediction is not tied to exact predicate strings observed during training, but generalizes across semantically equivalent formulations.

5 Conclusion

In this work, we formulated *3D spatial relation segmentation* in a feed-forward, pose-free multi-view setting, where the goal is to localize target objects that satisfy a spatial relation with respect to a visually specified subject using the underlying 3D scene structure. To tackle this, we proposed RelationVGGT, a feed-forward framework that integrates 3D geometry priors with 2D visual foundation models for subject-conditioned relational segmentation across views, without per-scene optimization or known camera poses. We also presented a fully automated data generation pipeline built on ScanNet++ with VLM and LLM integration, enabling scalable multi-view consistent annotations.

Limitations and future work. First, our data generation pipeline currently depends on datasets with instance-level annotations such as ScanNet++; extending it to unannotated or in-the-wild scenes is an important direction. Second, while we adopt Set-of-Mark prompting to extract relations via VLMs, exploring more accurate and efficient annotation strategies remains worthwhile. Third, our framework models object-object spatial relations, but could be extended to functional relations or combined with panoptic segmentation branches to jointly predict scene graph nodes and edges.

Acknowledgments and Disclosure of Funding

This work was supported by the Artificial Intelligence Graduate School Program, Yonsei University (No. RS-2020-II201361), the Institute of Information Communications Technology Planning Evaluation (IITP) grants funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea for the Global AI Frontier Lab International Collaborative Research (Nos. RS-2024-00469482 and RS-2024-00509279) and the Institute of Information communications Technology Planning evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-II220113, Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework)

References

[1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.

[3] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[4] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[5] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[6] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *NeurIPS*, 2024.

[7] Lojze Zust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. Panst3r: Multi-view consistent panoptic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5856–5866, 2025.

[8] Xiangyu Sun, Haoyi Jiang, Liu Liu, Seungtae Nam, Gyeongjin Kang, Xinjie Wang, Wei Sui, Zhizhong Su, Wenyu Liu, Xinggang Wang, et al. Uni3r: Unified 3d reconstruction and semantic understanding via generalizable gaussian splatting from unposed multi-view images. *arXiv preprint arXiv:2508.03643*, 2025.

[9] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[10] Sebastian Koch, Johanna Wald, Mirco Colosi, Narunas Vaskevicius, Pedro Hermosilla, Federico Tombari, and Timo Ropinski. Relationfield: Relate anything in radiance fields. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21706–21716, 2025.

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

[12] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[13] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[14] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[15] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[16] Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Tackling view-dependent semantics in 3d language gaussian splatting. *arXiv preprint arXiv:2505.24746*, 2025.

[17] Runsong Zhu, Ka-Hei Hui, Zhengzhe Liu, Qianyi Wu, Weiliang Tang, Shi Qiu, Pheng-Ann Heng, and Chi-Wing Fu. Cos3d: Collaborative open-vocabulary 3d segmentation. *arXiv preprint arXiv:2510.20238*, 2025.

[18] SungMin Jang and Wonjun Kim. Identity-aware language gaussian splatting for open-vocabulary 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20467–20476, 2025.

[19] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8350–8360, 2025.

[20] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing, 2017.

[21] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context, 2018.

[22] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts, 2018.

[23] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences, 2021.

[24] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions, 2020.

[25] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning, 2023.

[26] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9579–9589, 2024.

[27] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Z Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *Advances in Neural Information Processing Systems*, 37:6833–6859, 2024.

[28] Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, et al. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001*, 2025.

[29] Ye Liu, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. *arXiv preprint arXiv:2509.18094*, 2025.

[30] Shuting He, Guangquan Jie, Changshuo Wang, Yun Zhou, Shuming Hu, Guanbin Li, and Henghui Ding. Refersplat: Referring segmentation in 3d gaussian splatting. In *ICML*, 2025.

[31] Ozan Unal, Christos Sakaridis, Suman Saha, and Luc Van Gool. Four ways to improve verbo-visual fusion for dense 3d visual grounding, 2024.

[32] Xulun Ye, Qin Zhang, and Kun Zhou. Langref3dgs: Natural language-guided 3d referential segmentation from partial observations via 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38595–38605, June 2026.

[33] Changli Wu, Haodong Wang, Jiayi Ji, Yutian Yao, Chunsai Du, Jihua Kang, Yanwei Fu, and Liujuan Cao. Mvggt: Multimodal visual geometry grounded transformer for multiview 3d referring expression segmentation, 2026.

[34] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[35] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[37] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. In *arXiv preprint arXiv:2511.21631*, 2025.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[39] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023.

[40] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[41] Yoonwoo Jeong, Cheng Sun, Yu-Chiang Frank Wang, Minsu Cho, and Jaesung Choe. Mv-sam: Multi-view promptable segmentation using pointmap guidance, 2026.

A Dataset Construction and Quality

A.1 Relation distribution

Figure 5 shows the final predicate distribution in our training annotations. The distribution is strongly long-tailed: frequent relations such as *standing on*, *attached to*, and *supporting* coexist with a large number of low-frequency expressions, reflecting the lexical diversity of the open-vocabulary annotations and motivating our multi-query supervision rather than a fixed predicate classifier.

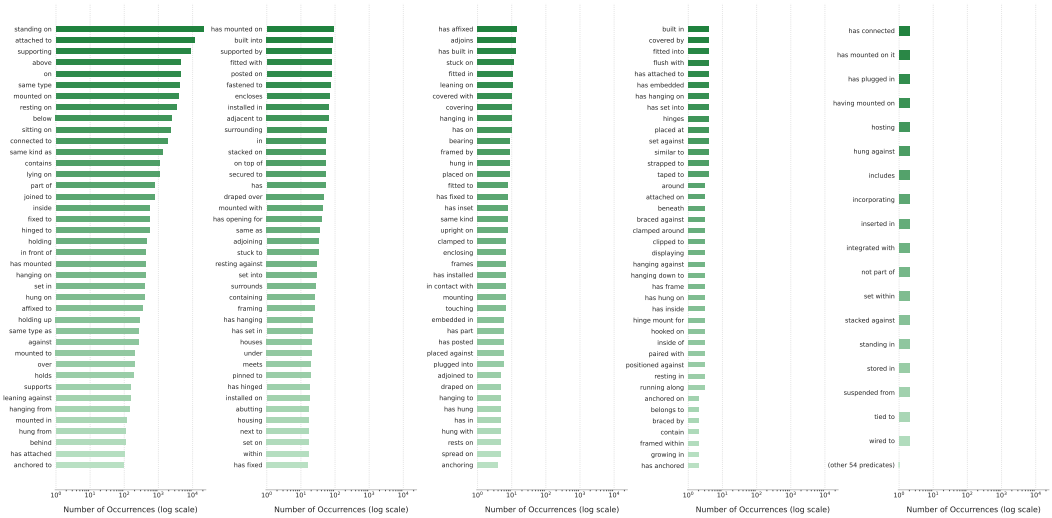

Figure 5: Distribution of the final relation predicates used for training. The horizontal axis shows the number of occurrences on a logarithmic scale; 54 additional low-frequency predicates are grouped for visualization.

A.2 Human audit of annotation quality

To quantify the quality of the automatically generated supervision, two authors independently audited all 4,695 directed relation proposals produced across 25 ScanNet++ scenes. For each proposal, an annotator was shown the relation proposition together with its supporting views, but not the pipeline's keep/remove decision, and assigned one of three labels: *correct*, *incorrect*, or *ambiguous*. We exclude ambiguous judgments from the denominators of the quality rates reported below. Since the audit covers a finite set of scenes, we compute 95% confidence intervals with a scene-level cluster bootstrap using 20,000 resamples. Unless otherwise noted, each reported rate is the mean of the two annotators.

Stage	Relations after stage	% of Stage 1
Stage 1: initial aggregation	4,695	100.0
Stage 2: filtering	770	16.4

Table 6: Number of relation proposals that survive the annotation filtering pipeline. (25 scenes as samples) Stage 2 combines multi-view consistency with the geometry-based filtering described in Sec. 3.5.

A.2.1 How many proposals survive filtering?

Table 6 summarizes the number of relation proposals before and after the multi-view and geometric filtering stage. Stage 2 removes 3,925 of the 4,695 aggregated proposals. This filtering is intentionally conservative: a relation supported by fewer than three frames is discarded even when it may be correct, because a false relation that survives the filter becomes direct training supervision, whereas a removed correct relation primarily reduces coverage.

Table 7: Human judgment versus the Stage-2 filtering decision. Counts are reported as annotator 1 / annotator 2; ambiguous judgments are excluded.

Human judgment	Kept by Stage 2	Removed by Stage 2	Total
Correct	506 / 421	1,151 / 1,173	1,657 / 1,594
Incorrect	257 / 198	2,748 / 2,494	3,005 / 2,692
Total	763 / 619	3,899 / 3,667	4,662 / 4,286

Table 8: Human evaluation of the annotation pipeline on 25 ScanNet++ scenes. Confidence intervals are obtained with scene-level cluster bootstrap using 20,000 resamples.

Statistic	Mean (%)	95% CI	Annotator 1 / 2 (%)
Precision after Stage 1	36.4	[32.2, 39.2]	35.5 / 37.2
Precision after Stage 2	**67.2**	**[60.3, 73.5]**	66.3 / 68.0
Incorrect proposals removed	**92.0**	**[89.1, 94.0]**	91.4 / 92.6
Correct proposals retained	28.5	[22.8, 37.2]	30.5 / 26.4

A.2.2 Human judgment versus filtering decision

Table 7 compares the human judgments with the Stage-2 keep/remove decision. Counts are reported as annotator 1 / annotator 2. The total numbers differ because ambiguous judgments are excluded from the table and all derived rates; the two annotators marked 33 and 409 proposals as ambiguous, respectively.

The filter removes the large majority of proposals judged incorrect, although it also discards a substantial number of correct proposals. The latter is expected from the conservative $K = 3$ multi-view agreement requirement, which rejects relations that are visible or confidently recognized in only one or two frames.

A.2.3 Annotation quality

We summarize annotation quality in Table 8. Precision after Stage 1 is the fraction of aggregated proposals judged correct, while precision after Stage 2 is the fraction of retained training relations judged correct. "Incorrect proposals removed" measures the fraction of human-judged incorrect proposals discarded by Stage 2, and "Correct proposals retained" measures the fraction of human-judged correct proposals that survive the filtering stage.

Filtering increases proposal precision from 36.4% to 67.2% and removes 92.0% of proposals judged incorrect. The 28.5% correct-retention rate should not be interpreted as recall over all true scene relations; it only measures retention within the Stage-1 proposal pool.

The two annotators agree on 83.1% of all proposals (Cohen's $\kappa = 0.670$), with a reported $\kappa = 0.816$ when ambiguous judgments are excluded. These results indicate that the filtering pipeline substantially improves the precision of the automatically generated supervision, while also showing that the retained set remains noisy and intentionally sacrifices coverage for reliability.

B Implementation Details

B.1 Training details

For each training sample, let $\mathcal{Q}$ denote the set of compatible phrasings generated from the annotated spatial relation. All queries $q \in \mathcal{Q}$ describe the same relation and are therefore supervised using the same ground-truth target mask M_t. Rather than treating these phrasings as mutually exclusive classes, we evaluate each query independently as a binary relation prediction.

For a target-view pixel u and query q, we denote the predicted probability by

$$p = S_t(u; q), \tag{14}$$

and the corresponding ground-truth label by

$$y = M_t(u) \in \{0, 1\}. \tag{15}$$

The focal loss for each pixel–query pair is

$$\ell_{\text{focal}}(p, y) = -\alpha_y (1 - p_y)^\gamma \log p_y, \tag{16}$$

where

$$p_y = \begin{cases} p, & y = 1, \\ 1 - p, & y = 0, \end{cases} \qquad \alpha_y = \begin{cases} \alpha, & y = 1, \\ 1 - \alpha, & y = 0. \end{cases} \tag{17}$$

We use $\alpha = 0.75$ and $\gamma = 1.5$ in all experiments.

Let Ω_t denote the set of pixels in the target view. The multi-query focal loss is obtained by averaging over both pixels and compatible queries:

$$\mathcal{L}_{\text{focal}}^{\text{multi-q}} = \frac{1}{|\mathcal{Q}|\,|\Omega_t|} \sum_{q \in \mathcal{Q}} \sum_{u \in \Omega_t} \ell_{\text{focal}}\left(S_t(u; q), M_t(u)\right). \tag{18}$$

Similarly, the Dice loss is computed independently for each compatible query against the same target mask and averaged over $\mathcal{Q}$:

$$\mathcal{L}_{\text{dice}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathcal{L}_{\text{dice}}\left(S_t(\cdot; q), M_t\right). \tag{19}$$

The overall mask supervision is

$$\mathcal{L}_{\text{mask}} = w_{\text{focal}} \mathcal{L}_{\text{focal}}^{\text{multi-q}} + w_{\text{dice}} \mathcal{L}_{\text{dice}}. \tag{20}$$

This formulation allows multiple semantically valid descriptions of the same spatial relation to serve as positive supervision without forcing competition between them. Further implementation details are available in our released code.

B.2 Inference efficiency

Table 9 compares inference efficiency with RelationField, which requires scene-specific optimization before relation queries can be evaluated. On an NVIDIA L40S, our method requires 26 ms for inference in the timing setup used in our evaluation, whereas RelationField requires 288 s for a relation query and more than 3 hours of per-scene optimization.

The timing comparison reflects the different operating regimes of the two methods: RelationField constructs a scene-specific representation, while our model performs feed-forward inference on unseen scenes without scene-level optimization.

Method	Query inference	Per-scene optimization
RelationField	288 s	>3 h
Ours	26 ms	None

Table 9: Inference-time comparison with RelationField on an NVIDIA L40S. RelationField additionally requires scene-specific optimization before query-time inference.

B.3 Annotation prompts

Figures 6 and 7 show the exact prompts used in our annotation pipeline. For visual relation extraction, the VLM receives a single SoM-tagged image and is instructed to identify only visible tagged instances and return directed, open-vocabulary relations in a structured JSON format. The prompt explicitly discourages unsupported or ambiguous relations, preserves the subject→object direction, and provides relation families only as references rather than as a closed predicate vocabulary.

The subsequent relation-refinement stage is deliberately restricted to text-only, category-level information. Given the subject category, input relation, object category, and relation family, the LLM is asked to produce a short, natural predicate phrase while preserving both the original semantic intent and the relation family. It receives no image or scene geometry and is explicitly prohibited from inferring instance-level pose or layout, which separates lexical refinement from the visual evidence used to establish the relation itself.

[System] You are given one SoM-tagged indoor scene image. Identify visible tagged objects and infer visually grounded open-vocabulary relationships between them.

Input. One indoor scene image in which each visible object instance has exactly one integer tag placed near its segmented region.

Output. Return strict JSON only with two fields: `objects` and `relationships`. `objects` maps each visible `tag_id` to a short singular object name. `relationships` is a list of `{s_id, o_id, predicates}` entries.

Object rules. Use only `visible tagged instances`. Ignore untagged, hidden, guessed, or heavily ambiguous objects. The SoM image is the primary evidence source; tags are grounding aids for object identity.

Relationship rules. Infer only visually supported relations. Do not force a relation for every pair. Merge valid non-redundant predicates for the same ordered pair. Predicates are open-vocabulary, and the listed examples are only references, not a fixed label set.

Core constraints. Avoid vague predicates such as `related_to` or `associated_with`. Do not use `left`, `right`, `in_front_of`, or `behind`. Preserve subject→object direction. If evidence is weak, blurry, occluded, or ambiguous, omit the relation.

Relation families.

1. composition	`part_of`, `component_of`
2. topology	`inside`, `contains`, `attached_to`
3. support-contact	`on`, `resting_on`, `mounted_on`
4. vertical directional	`above`, `below`
5. proximity/adjacency	`near`, `adjacent_to`, `next_to`
6. comparative/similarity	`same_type_as`, `same_color_as`, `larger_than`

Example output.

```
{
  "objects": { "4": "floor", "7": "table" },
  "relationships": [
    { "s_id": "7", "o_id": "4", "predicates": ["standing_on"] }
  ]
}
```

Figure 6: Prompt for grounded open-vocabulary relationship extraction.

[System] You are a commonsense relation rephraser for object-category triples.

Input. A structured triple with four fields: `subject_category`, `relation`, `object_category`, and `relation_family`.

Task. Refine the relation so that it sounds more natural, idiomatic, and category-appropriate, using only `category-level commonsense`.

Scope. Text-only task. No image, no scene geometry, and no instance-level evidence. Do not infer pose, layout, attachment, or visibility.

Core objective. Improve awkward or unidiomatic wording while preserving the original semantic intent. Preserve `relation_family` exactly. Make only conservative, same-family refinements, and avoid hallucinated edits.

Relation-family rule. The input relation belongs to one fixed family. Improve wording only within that family. Never cross families, even if another family seems plausible.

Relation families.

1. composition	part-whole, component, membership.
2. topology	containment, enclosure, attachment.
3. support-contact	resting, hanging, leaning, mounting.
4. vertical directional	vertical ordering only.
5. proximity/adjacency	local closeness.
6. comparative/similarity	size, color, shape, material.

Examples.

`(TV, hanging_on, wall, support-contact)` → `mounted_on`
`(pillow, standing_on, bed, support-contact)` → `lying_on`
`(chair, located_at, table, proximity/adjacency)` → `next_to`

Lexical constraints. `refined_relation` must be a short predicate-style phrase (e.g., `mounted_on`, `lying_on`, `next_to`, `similar_to`). Do not include the subject or object noun. Avoid clauses such as *"attached to the wall"*. Preserve subject→object direction.

Output (JSON only).

```
{
  "subject_category": "<subject_category>",
  "input_relation": "<original_relation>",
  "refined_relation": "<same-family naturalized relation>",
  "object_category": "<object_category>",
  "relation_family": "<same as input>",
  "changed": <true | false>,
  "confidence": "<high | medium | low>"
}
```

Figure 7: Prompt for relation refinement.

Method	Subject specification	Target category in input	Test-scene optimization	Avg. mIoU / mAcc@0.25
MVGGT	Text	Yes	No	0.153 / 0.217
ReferSplat	Text	No	Yes	0.213 / 0.263
Ours	Visual mask	No	No	**0.382 / 0.549**

Table 10: Comparison with closely related 3D referring-segmentation methods. The methods differ in their native input interfaces and optimization protocols; in particular, MVGGT receives the target category in the evaluated text template, whereas ReferSplat and our method do not.

		Ours		RelationField	
Target-view subset	Views	mIoU	mAcc	mIoU	mAcc
Subject visible	257	0.440	0.677	0.318	0.510
Subject not visible	23	0.507	0.652	0.159	0.217

Table 11: Performance on ScanNet++ target views grouped by subject visibility. The subject is always specified in the reference view; visibility here refers only to the evaluated target view.

C Additional Experiments

We report three analyses that complement the results in the main paper: comparisons with closely related 3D referring-segmentation methods, robustness when the reference subject is not visible in a target view, and controls in which the predicate meaning is deliberately changed. Experiments already reported in the main paper, including subject-mask specification and meaning-preserving predicate rephrasings, are not repeated here.

C.1 Comparison with 3D referring-segmentation methods

We additionally compare with MVGGT [33] and ReferSplat [30], two closely related 3D referring-segmentation methods. Their interfaces differ from ours: neither method accepts a visual subject mask, so the subject must instead be specified through language. We therefore adapt each method using its native text-query interface.

MVGGT. We evaluate the released checkpoint zero-shot. Each benchmark triplet $(\text{subject}, \text{relation}, \text{target})$ is converted into a ScanRefer-style referring expression, following the linguistic form used to train MVGGT. For example, $(\text{backpack}, \text{leaning against}, \text{wall})$ is expressed as "this is a wall. it has the backpack leaning against it." We evaluate six query templates and report the best-performing template. In this adaptation, the target category is explicitly named in the text.

ReferSplat. ReferSplat requires per-scene optimization, so we fit its Gaussian representation on each benchmark scene and train its referring branch using the method's native pipeline. At evaluation time, we hide the target category by replacing it with a generic noun, e.g., "the object that the cushion is lying on." Thus, the query specifies the subject and relation while withholding the target category.

As shown in Table 10, our method achieves substantially higher average segmentation accuracy than the adapted referring-segmentation baselines. A key difference lies in how the reference subject is specified. In text-only referring, the subject must itself be grounded from a noun phrase, which can be ambiguous when multiple instances of the same category are present in the scene. Our formulation instead visually specifies the subject, explicitly resolving this instance-level ambiguity and allowing the model to focus on identifying the target from the relation. This comparison is not a matched-input ablation, since the methods also differ in query form, pretraining data, and test-scene optimization; nevertheless, the results highlight the practical advantage of separating subject grounding from relation-conditioned target localization.

C.2 Subject visibility across target views

The subject is specified only in the reference view and may not be visible in every target view. To evaluate this case explicitly, we label every scored target view in the ScanNet++ benchmark by whether the subject is visible and re-score both our method and RelationField on the two subsets.

Our method remains effective when the subject is absent from the evaluated target view. This behavior is consistent with the model design: the subject is grounded in the reference view, while cross-view attention allows target-view tokens to access subject-conditioned evidence from the multi-view set. We do not interpret the numerically higher mIoU on the not-visible subset as an intrinsic advantage, since this subset contains only 23 views and may differ in target size and difficulty.

Table 12: Meaning-changing predicate controls. ΔmIoU is measured relative to the original benchmark predicate while keeping the original target mask fixed. These controls therefore measure query sensitivity rather than segmentation accuracy for the substituted relation itself.

Predicate substitution	n	ΔmIoU
Random relation from an incompatible family	119	-0.158 $[-0.207, -0.111]$
Relation that changes the correct target	118	-0.276

C.3 Sensitivity to meaning-changing predicate controls

The main paper evaluates meaning-preserving predicate rephrasings. Here, we provide a complementary control in which the predicate meaning is deliberately changed while the scene, subject mask, and original target annotation are held fixed. This tests whether the prediction is actually sensitive to the relational text rather than simply reproducing a target independent of the query. The results are summarized in Table 12.

Replacing the predicate with a random relation from an incompatible family reduces mIoU by 0.158. When the substituted relation is selected so that the correct target changes, the degradation increases to 0.276. In contrast to the meaning-preserving substitutions reported in the main paper, these meaning-changing controls substantially alter the prediction. This supports that the model responds to the semantic content of the relation query rather than only to the visual subject or a fixed relation label.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We claim the main contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss them at conclusion sections.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We provide proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We will provide open-source codes.

   Guidelines:

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

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [N/A]

   Justification: We will provide it later.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes it does at experiments section.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: Not related well.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: About GPU hours.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Nothing to violate in this work.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: At Introduction section.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Not related.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We detailed at main section.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: Not related.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Never. not related.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: To polish writing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.