LangRef3DGS: Natural Language-Guided 3D Referential Segmentation from Partial Observations via 3D Gaussian Splatting

Supplementary Material

A. Derivation of Dirichlet Process for Novel Class Discovery

This appendix provides the complete variational inference derivation for the Dirichlet Process Gaussian Mixture Model (DP-GMM) used in Section 4.2 of the main paper for novel class discovery. The derivation follows the standard mean-field variational inference framework for DP mixtures , adapted to our semantic feature space.

A.1. DP-GMM Formulation over Semantic Features

Let $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]^\top \in \mathbb{R}^{N \times d}$ denote the semantic feature matrix of N 3D Gaussians. We model the distribution of these features using a DP-GMM:

$$p(\mathbf{f}_i) = \sum_{k=1}^{K^*} \pi_k \mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{A.1}$$

where K^* denotes the (potentially infinite) number of mixture components, π_k are the mixing weights, and $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ represent the mean and covariance parameters of component k.

The mixing weights $\{\pi_k\}_{k=1}^{\infty}$ are generated through the stick-breaking construction:

$$\pi_k = v_k \prod_{j<k} (1 - v_j), \quad v_k \sim \mathrm{Beta}(1, \alpha), \tag{A.2}$$

where $\alpha > 0$ is the concentration parameter controlling the prior tendency to create new clusters.

The complete generative process can be expressed hierarchically as:

$$G \mid G_0, \alpha \sim \mathrm{DP}(G_0, \alpha), \tag{A.3}$$
$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mid G \sim G, \quad k = 1, 2, \ldots, \infty, \tag{A.4}$$
$$z_i \mid \{\pi_k\}_{k=1}^{\infty} \sim \mathrm{Categorical}(\{\pi_k\}_{k=1}^{\infty}), \tag{A.5}$$
$$\mathbf{f}_i \mid z_i = k \sim \mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad i = 1, \ldots, N, \tag{A.6}$$

where G_0 is the base measure (we use a Normal-Inverse-Wishart distribution for conjugacy), and $z_i \in \{1, 2, \ldots\}$ is the latent cluster assignment for feature vector $\mathbf{f}_i$.

For practical implementation, we truncate the infinite mixture at a maximum of K components, which is valid when K is sufficiently large relative to the expected number of clusters in the data .

A.2. Evidence Lower Bound (ELBO)

Given the observed semantic features $\mathbf{F}$, we aim to approximate the intractable posterior $p(\mathbf{V}, \boldsymbol{\theta}^*, \mathbf{Z} \mid \mathbf{F})$, where:

- $\mathbf{V} = \{v_1, \ldots, v_{K-1}\}$ are the stick-breaking variables,
- $\boldsymbol{\theta}^* = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ are component parameters,
- $\mathbf{Z} = \{z_i\}_{i=1}^{N}$ are cluster assignments.

We introduce a factorized variational posterior distribution:

$$q(\mathbf{V}, \boldsymbol{\theta}^*, \mathbf{Z}) = \prod_{k=1}^{K-1} q_{\gamma_k}(v_k) \prod_{k=1}^{K} q_{\tau_k}(\boldsymbol{\theta}_k^*) \prod_{i=1}^{N} q_{\phi_i}(z_i), \tag{A.7}$$

with the following variational families:

$$q_{\gamma_k}(v_k) = \mathrm{Beta}(v_k \mid \gamma_{k,1}, \gamma_{k,2}), \tag{A.8}$$
$$q_{\tau_k}(\boldsymbol{\theta}_k^*) = \mathrm{NIW}(\boldsymbol{\theta}_k^* \mid \mathbf{u}_k, c_k, \mathbf{B}_k, \nu_k), \tag{A.9}$$
$$q_{\phi_i}(z_i) = \mathrm{Categorical}(z_i \mid \boldsymbol{\phi}_i), \tag{A.10}$$

where $\phi_{i,k} = q(z_i = k)$ represents the responsibility of component k for feature $\mathbf{f}_i$.

The variational inference objective is to maximize the Evidence Lower Bound (ELBO):

$$\begin{aligned} \log p(\mathbf{F} \mid \alpha, \lambda) &\geq \mathcal{L}(\gamma, \tau, \phi) \\ &= \mathbb{E}_q[\log p(\mathbf{V} \mid \alpha)] + \mathbb{E}_q[\log p(\boldsymbol{\theta}^* \mid \lambda)] \\ &\quad + \sum_{i=1}^{N} \mathbb{E}_q[\log p(z_i \mid \mathbf{V})] \\ &\quad + \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathbf{f}_i \mid z_i, \boldsymbol{\theta}^*)] \\ &\quad - \mathbb{E}_q[\log q(\mathbf{V}, \boldsymbol{\theta}^*, \mathbf{Z})]. \end{aligned} \tag{A.11}$$

Expanding each term and collecting relevant expectations, we obtain:

$$\mathcal{L}(\gamma, \tau, \phi) = \sum_{k=1}^{K-1} \mathbb{E}_q[\log \mathrm{Beta}(v_k \mid 1, \alpha)] + \sum_{k=1}^{K} \mathbb{E}_q[\log \mathrm{NIW}(\boldsymbol{\theta}_k^* \mid \lambda)] + \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{i,k} \left(\mathbb{E}_q[\log v_k] + \sum_{j<k} \mathbb{E}_q[\log(1 - v_j)] \right) + \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{i,k}\, \mathbb{E}_q[\log \mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] - \sum_{k=1}^{K-1} \mathbb{E}_q[\log q_{\gamma_k}(v_k)] - \sum_{k=1}^{K} \mathbb{E}_q[\log q_{\tau_k}(\boldsymbol{\theta}_k^*)] - \sum_{i=1}^{N} \mathbb{E}_q[\log q_{\phi_i}(z_i)]. \tag{A.12}$$

A.3. Coordinate Ascent Updates

The optimal variational parameters are obtained by coordinate ascent on the ELBO. We derive each update in closed form:

Update for $\phi_{i,k}$: The responsibility of component k for feature $\mathbf{f}_i$ is updated as:

$$\log \phi_{i,k} \propto \underbrace{\Psi(\gamma_{k,1}) - \Psi(\gamma_{k,1} + \gamma_{k,2})}_{\mathbb{E}_q[\log v_k]} + \underbrace{\sum_{j<k} \Big(\Psi(\gamma_{j,2}) - \Psi(\gamma_{j,1} + \gamma_{j,2}) \Big)}_{\sum_{j<k} \mathbb{E}_q[\log(1-v_j)]} + \underbrace{\mathbb{E}_q[\log \mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]}_{\text{data likelihood}}. \tag{A.13}$$

where $\Psi(\cdot)$ denotes the digamma function, and the Gaussian term expectations are computed as:

$$\mathbb{E}_q[\log \mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] = \frac{1}{2}\mathbb{E}_q[\log |\boldsymbol{\Sigma}_k^{-1}|] - \frac{d}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_q[(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{f}_i - \boldsymbol{\mu}_k)]. \tag{A.14}$$

$$\mathbb{E}_q[\log |\boldsymbol{\Sigma}_k^{-1}|] = \sum_{t=1}^{d} \psi\left(\frac{\nu_k + 1 - t}{2}\right) + d\log 2 + \log |\mathbf{B}_k|. \tag{A.15}$$

$$\mathbb{E}_q[(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{f}_i - \boldsymbol{\mu}_k)] = \nu_k(\mathbf{f}_i - \mathbf{u}_k)^\top \mathbf{B}_k(\mathbf{f}_i - \mathbf{u}_k) + \mathrm{Tr}(\mathbf{B}_k \tilde{\boldsymbol{\Sigma}}_i). \tag{A.16}$$

where $\tau_k = \{\mathbf{u}_k, c_k, \mathbf{B}_k, \nu_k\}$ are the variational parameters for component k.

After computing the unnormalized values, we apply the normalization $\phi_{i,k} \leftarrow \frac{\exp(\log \phi_{i,k})}{\sum_{j=1}^{K} \exp(\log \phi_{i,j})}$.

Update for γ_k: The variational Beta parameters for stick-breaking variables are:

$$\gamma_{k,1} = 1 + \sum_{i=1}^{N} \phi_{i,k}, \tag{A.17}$$

$$\gamma_{k,2} = \alpha + \sum_{i=1}^{N} \sum_{j>k} \phi_{i,j}, \tag{A.18}$$

Update for component parameters $\tau_k = \{\mathbf{u}_k, c_k, \mathbf{B}_k, \nu_k\}$:

$$c_k = c_0 + N_k, \tag{A.19}$$

$$\mathbf{u}_k = \frac{1}{c_k}\left(c_0 \mathbf{u}_0 + \sum_{i=1}^{N} \phi_{i,k}\mathbf{f}_i\right), \tag{A.20}$$

$$\nu_k = \nu_0 + N_k, \tag{A.21}$$

$$\mathbf{B}_k^{-1} = \mathbf{B}_0^{-1} + \sum_{i=1}^{N} \phi_{i,k}(\mathbf{f}_i - \mathbf{u}_k)(\mathbf{f}_i - \mathbf{u}_k)^\top + \sum_{i=1}^{N} \phi_{i,k}\tilde{\boldsymbol{\Sigma}}_i, \tag{A.22}$$

where $N_k = \sum_{i=1}^{N} \phi_{i,k}$ is the expected number of points assigned to component k.

A.4. ELBO-based Validation for Novel Classes

As described in Section 4.2 of the main paper, a semantic feature $\mathbf{f}_{\text{new}}$ becomes a candidate for a new class when it lies in a low-density region of the existing mixture:

$$\max_k \mathcal{N}(\mathbf{f}_{\text{new}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) < \varepsilon. \tag{A.23}$$

To prevent spurious cluster creation, we evaluate the change in ELBO that would result from adding a new component $K+1$ initialized at $\mathbf{f}_{\text{new}}$. The approximate ELBO gain is:

$$\Delta\mathrm{ELBO} = \mathcal{L}_{\text{new}} - \mathcal{L}_{\text{exist}} \approx \underbrace{\log \frac{\alpha}{N + \alpha}}_{\text{prior for new component}} + \underbrace{\log \mathcal{N}(\mathbf{f}_{\text{new}} \mid \boldsymbol{\mu}_{m^*}, \boldsymbol{\Sigma}_{m^*})}_{\text{data fit}} - \underbrace{\mathrm{KL}\big(q(\theta_{K+1}) \,\|\, p(\theta_{K+1})\big)}_{\text{complexity penalty}}. \tag{A.24}$$

where $m^* = \arg\max_k \mathcal{N}(\mathbf{f}_{\text{new}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the best-fit existing component used for initialization.

The KL divergence term penalizes the addition of unnecessary components:

$$\begin{aligned}\mathrm{KL}\big(q(\theta_{K+1}) \,\|\, p(\theta_{K+1})\big) &= \mathrm{KL}\big(\mathrm{NIW}(\cdot \mid \tau_{K+1}) \,\|\, \mathrm{NIW}(\cdot \mid \lambda)\big) \\ &= \frac{1}{2}\,\mathrm{Tr}(\mathbf{B}_{K+1}^{-1}\mathbf{B}_0) \\ &+ \frac{c_0}{2}(\mathbf{u}_{K+1}-\mathbf{u}_0)^\top \mathbf{B}_{K+1}(\mathbf{u}_{K+1}-\mathbf{u}_0) \\ &- \frac{\nu_{K+1}}{2}\log|\mathbf{B}_{K+1}| \\ &+ \frac{d}{2}\log\frac{c_{K+1}}{c_0} \\ &+ \log\frac{\Gamma_d(\nu_0/2)}{\Gamma_d(\nu_{K+1}/2)} \\ &+ \frac{\nu_{K+1}-\nu_0}{2}\,\psi_d\Big(\frac{\nu_{K+1}}{2}\Big) + \text{const.}\end{aligned} \tag{A.25}$$

where $\Gamma_d(\cdot)$ and $\psi_d(\cdot)$ are the multivariate gamma and digamma functions, respectively.

A new component is instantiated only if $\Delta\text{ELBO} > 0$, ensuring that only statistically justified clusters are added.

A.5. DP Posterior as Pseudo-labels

After validation, the DP posterior provides soft cluster assignments that serve as pseudo-labels for downstream optimization. The assignment probability for feature $\mathbf{f}_i$ to component k is:

$$q(z_i = k) = \phi_{i,k}, \tag{A.26}$$

where $\phi_{i,k}$ are the optimized responsibilities from Eq. (A.13).

The DP loss used to supervise the semantic feature learning is:

$$\mathcal{L}_{\text{DP}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} q(z_i = k)\log\mathcal{N}(\mathbf{f}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{A.27}$$

This loss encourages semantic features to concentrate around their assigned cluster centers while allowing the model to discover new categories beyond the predefined vocabulary. The gradient of this loss propagates through the entire 3DGS pipeline, jointly optimizing both the Gaussian parameters and the semantic embeddings.

B. Derivation of Gradient Low-Rank Mechanism

This appendix provides detailed mathematical derivations for the Gradient Low-Rank mechanism introduced in Section 4.3 of the main paper.

B.1. Gradient Form and Update Dynamics

Let $\mathbf{F}_t \in \mathbb{R}^{N\times d}$ denote the semantic feature matrix of N Gaussian points at training step t, and $\nabla_{\mathbf{F}}\mathcal{L}_t$ be the gradient matrix of the loss function $\mathcal{L}$ with respect to $\mathbf{F}_t$. We assume the gradient takes the following parametric form:

$$\nabla_{\mathbf{F}}\mathcal{L}_t = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_t\mathbf{C}_i) \tag{B.1}$$

Under vanilla SGD with learning rate η, the feature update rule is:

$$\mathbf{F}_{t+1} = \mathbf{F}_t + \eta\nabla_{\mathbf{F}}\mathcal{L}_t \tag{B.2}$$

Vectorizing $\mathbf{F}_t$ as $\mathbf{f}_t = \mathrm{vec}(\mathbf{F}_t) \in \mathbb{R}^{Nd}$ and using $\mathrm{vec}(\mathbf{BFC}) = (\mathbf{C}^\top \otimes \mathbf{B})\mathrm{vec}(\mathbf{F})$ yields:

$$\mathbf{g}_t = \mathbf{a}_t - \mathbf{S}\mathbf{f}_t \tag{B.3}$$

where $\mathbf{g}_t = \mathrm{vec}(\nabla_{\mathbf{F}}\mathcal{L}_t)$, $\mathbf{a}_t = \frac{1}{N}\sum_{i=1}^{N}\mathrm{vec}(\mathbf{A}_i)$, and $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{C}_i^\top \otimes \mathbf{B}_i$ is PSD.

The recursive relation becomes:

$$\mathbf{g}_{t+1} = (\mathbf{I} - \eta\mathbf{S})\mathbf{g}_t \tag{B.4}$$

B.2. Stable Rank Bound

The stable rank is defined as $\mathrm{sr}(\mathbf{M}) = \|\mathbf{M}\|_F^2/\|\mathbf{M}\|_2^2$.

Let $\lambda_1 < \lambda_2$ be the two smallest distinct eigenvalues of $\mathbf{S}$, with λ_1 having multiplicity κ_1. Let V_1 denote the minimal eigenspace. Decompose:

$$\mathbf{g}_{t_0} = \mathbf{g}_{t_0}^{\|} + \mathbf{g}_{t_0}^{\perp} \tag{B.5}$$

Since V_1 is invariant:

$$\mathbf{g}_t = (\mathbf{I}-\eta\mathbf{S})^{t-t_0}\mathbf{g}_{t_0}^{\|} + (\mathbf{I}-\eta\mathbf{S})^{t-t_0}\mathbf{g}_{t_0}^{\perp} \tag{B.6}$$

Spectral norm inequality:

$$\|\mathbf{g}_t\|_2^2 \le (1-\eta\lambda_1)^{2(t-t_0)}\|\mathbf{g}_{t_0}^{\|}\|_2^2 + (1-\eta\lambda_2)^{2(t-t_0)}\|\mathbf{g}_{t_0}^{\perp}\|_2^2 \tag{B.7}$$

Frobenius norm inequality:

$$\|\mathbf{g}_t\|_F^2 \ge (1-\eta\lambda_1)^{2(t-t_0)}\|\mathbf{g}_{t_0}^{\|}\|_F^2 \tag{B.8}$$

Combining yields:

$$\mathrm{sr}(\nabla_{\mathbf{F}}\mathcal{L}_t) \le \mathrm{sr}\Big((\nabla_{\mathbf{F}}\mathcal{L}_{t_0})^{\|}\Big) + \left(\frac{1-\eta\lambda_2}{1-\eta\lambda_1}\right)^{2(t-t_0)}\frac{\|(\nabla_{\mathbf{F}}\mathcal{L}_{t_0})^{\perp}\|_F^2}{\|\nabla_{\mathbf{F}}\mathcal{L}_{t_0}\|_2^2} \tag{B.9}$$

B.3. Low-Rank Structure of Semantic Gradients

When the gradient has structure $\nabla_{\mathbf{F}}\mathcal{L}_t = \frac{1}{N}\sum_i(\mathbf{a}_i - \mathbf{B}_i\mathbf{F}_t\mathbf{f}_i^\top)$ with $\mathrm{rank}(\{\mathbf{f}_i\}) = N' < d$:

$$\mathrm{sr}\left((\nabla_{\mathbf{F}}\mathcal{L}_{t_0})^{\|}\right) \leq d - N' \tag{B.10}$$

Thus for large t:

$$\mathrm{sr}(\nabla_{\mathbf{F}}\mathcal{L}_t) \leq \min(d - N',\, N') \leq \frac{d}{2} \tag{B.11}$$

B.4. Gradient Low-Rank Projection

Using truncated SVD:

$$\nabla_{\mathbf{F}}\mathcal{L}_t \approx \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^\top = \sum_{i=1}^{r} s_i\mathbf{u}_i\mathbf{v}_i^\top \tag{B.12}$$

Projected gradient:

$$\tilde{\nabla}_{\mathbf{F}}\mathcal{L}_t = \mathbf{P}_t^\top(\nabla_{\mathbf{F}}\mathcal{L}_t)\mathbf{Q}_t \tag{B.13}$$

Projected feature update:

$$\mathbf{F}_{t+1} = \mathbf{F}_t - \eta\mathbf{P}_t\tilde{\nabla}_{\mathbf{F}}\mathcal{L}_t\mathbf{Q}_t^\top \tag{B.14}$$

B.5. Convergence Guarantee

The projected gradient $\mathbf{R}_t = \mathbf{P}^\top(\nabla_{\mathbf{F}}\mathcal{L}_t)\mathbf{Q}$ satisfies:

$$\|\mathbf{R}_t\|_F \leq \left[1 - \eta\left(\kappa - L_A - L_B L_C D^2\right)\right]\|\mathbf{R}_{t-1}\|_F \tag{B.15}$$

This establishes linear convergence when $\kappa > L_A + L_B L_C D^2$, validating our low-rank approach.

C. Implementation Details

In this section, we provide comprehensive technical specifications of our framework to ensure reproducibility.

C.1. Architecture and Encoders

The proposed **LangGS** (Language-Guided 3DGS) framework utilizes the following components:

- **Encoders**: We employ a pre-trained **CLIP (ViT-B/16)** model to extract aligned text and image embeddings, which serve as the foundation for our open-vocabulary semantic field.
- **Prototypes**: Semantic centroids (prototypes) are initialized using CLIP embeddings. These are further refined by aggregating 3D Gaussian features assigned to specific clusters by the Dirichlet Process (DP) module, ensuring they adapt to the geometric constraints of the **partial observations**.
- **Feature Lifting**: To project 2D semantics into 3D space, semantic features are rendered via alpha-blending. We supervise this process using pseudo-labels generated by **Grounded-SAM**, optimized through a cross-entropy loss function.

C.2. Gradient Low-Rank (GLR) Mechanism

The efficiency of the GLR mechanism is achieved through several strategic optimizations:

- **Computational Efficiency**: To minimize overhead, Singular Value Decomposition (SVD) is performed on the $d \times d$ ($d = 512$) covariance matrix rather than the high-dimensional $N \times d$ feature matrix. This significantly reduces the computational complexity.
- **Hyperparameters**: We empirically set the rank to $r = 16$ with an update frequency of 100 iterations.
- **Optimization**: Extensive ablation studies on the rank r and cluster count K demonstrate an optimal balance between feature discriminability and processing speed. The complete sensitivity analysis and source code will be made available to the community.

C.3. Runtime and Efficiency Analysis

We evaluate the runtime efficiency on the **Ramen scene** using an NVIDIA RTX 4080 GPU. As summarized in Table 6, our framework retains the high-speed advantage of 3D Gaussian Splatting due to two factors: (1) the **DP module** is triggered periodically rather than at every iteration; (2) the **GLR mechanism** operates on a compact covariance matrix, incurring negligible latency. These results substantiate the real-time capability of our method for both training and inference, even under **sparse-view** conditions.

Table 6. Efficiency comparison on the Ramen scene. Our method achieves superior inference speed and training efficiency compared to state-of-the-art baselines.

Method	Inf. (FPS) ↑	Train (Time) ↓	GPU Mem. ↓
LangSplat [34]	102	80 min	4.0 GB
Gaussian Grouping [48]	170	66 min	10.4 GB
Ours (LangGS)	**185**	**52 min**	**6.8 GB**

C.4. RGB-D and Depth Usage

Following SOTA (e.g., LERF, LangSplat), we use monocular (e.g., Depth-Anything) or SfM-derived depth when hardware depth is absent. These priors constrain Gaussian positions and supervise depth loss, ensuring geometric consistency under sparse views. We will refine our descriptions in the revised manuscript to ensure clarity and prevent misinterpretation.

D. Dense-view Ablation Studies

As demonstrated in the table below, we conduct additional ablation experiments under the dense-view setting

to evaluate the individual contributions of our core components. The results confirm that both the Gradient Low-Rank (GLR) mechanism and the Dirichlet Process (DP) module are essential for achieving optimal performance. Specifically, removing either GLR or DP leads to a noticeable drop in mIoU, whereas the full model achieves the highest score of 60.69, validating the synergy between adaptive clustering and semantic refinement in fully observed 3D scenes.

Complete-view	Ours w/o GLR	Ours w/o DP	Ours (Full)
mIoU ↑	56.67	57.63	**60.69**

E. Failure Cases and Scale

Failure Cases and Scale Analysis. While our framework achieves robust 3D segmentation, it still faces challenges in certain edge cases. Primary failure cases occur in scenes with extreme scale variations, where very small objects may be over-smoothed by the Gaussian-based semantic embeddings. Additionally, under severe partial observations(e.g., visibility below 20%), the Dirichlet Process (DP) may occasionally over-cluster fragmented regions into a single novel category due to insufficient geometric cues and low density of visual features. These scale-related limitations and boundary ambiguities suggest that incorporating multi-scale feature hierarchies or hierarchical DP priors could be a promising direction for future refinement of open-vocabulary 3D scene understanding.

F. Additional Qualitative Results

To further demonstrate the robustness and generalization of our approach across diverse environments, we provide extended qualitative evaluations.

F.1. Scene Diversity and Complexity

As shown in Fig. 6, Fig. 7, and Fig. 8, we visualize results on three representative tabletop scenes. These examples encompass a wide range of spatial layouts and object densities, highlighting our model's ability to interpret complex **language-guided instructions**. The visualizations underscore how our method consistently produces precise, well-localized segmentations even when objects exhibit significant **occlusions** or are surrounded by visually similar distractors.

F.2. Large-Scale and Real-World Generalization

To evaluate the scalability of our framework, we extend our experiments to larger-scale and more challenging datasets:

- **Mip-360 (Garden Scene)**: Fig. 5 presents a comparative analysis in the *Garden* scene from the Mip-360 dataset. Despite the **sparse-view** nature and complex outdoor background, our method maintains high semantic consistency.
- **ScanNet Subset**: In Fig. 4, we show qualitative comparisons on the ScanNet subset, which represents typical indoor room-scale environments. Our approach effectively handles **partial observations** common in handheld RGB-D scans, outperforming baseline methods in maintaining sharp object boundaries under cluttered conditions.

These additional results further substantiate that our **LangGS** framework is not only robust to **partial inputs** but also highly generalizable across different sensor types and scene scales.

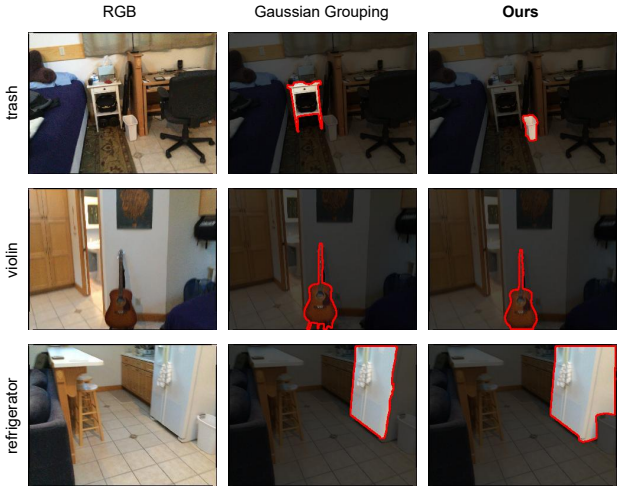

Figure 4. 3D segmentation in the *scannet* scene.

Figure 5. 3D segmentation in the *Mips-360* scene.

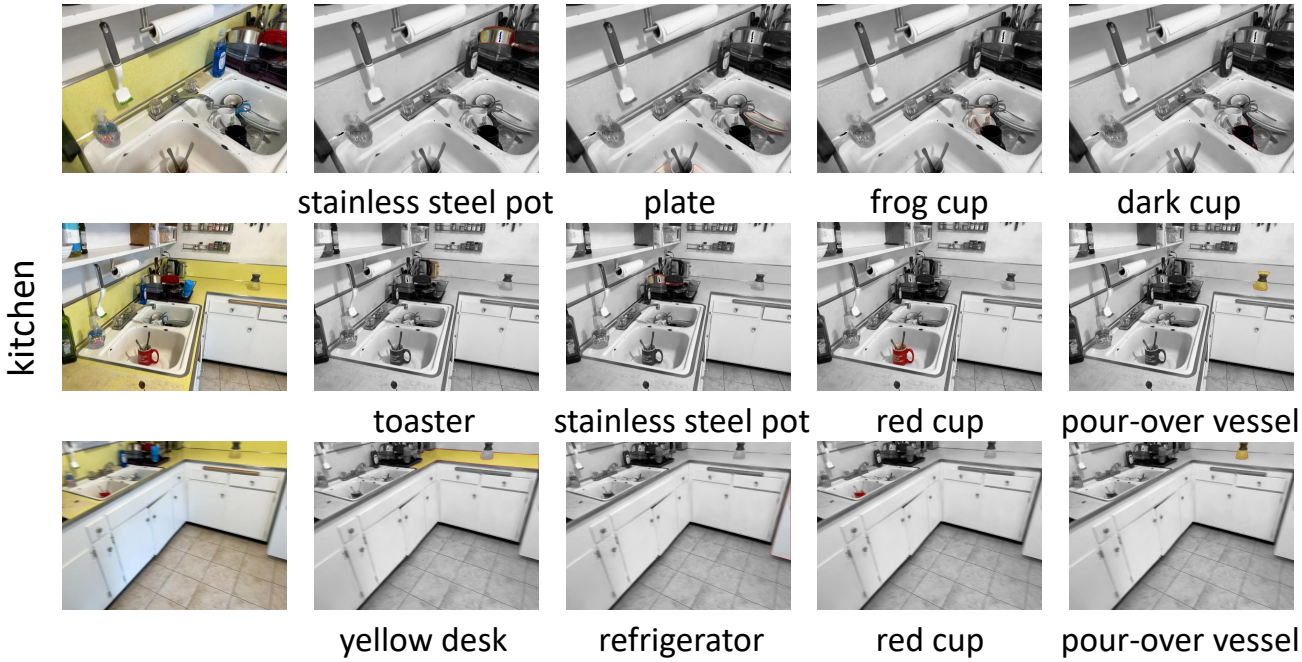

Figure 6. 3D segmentation in the *kitchen* scene.

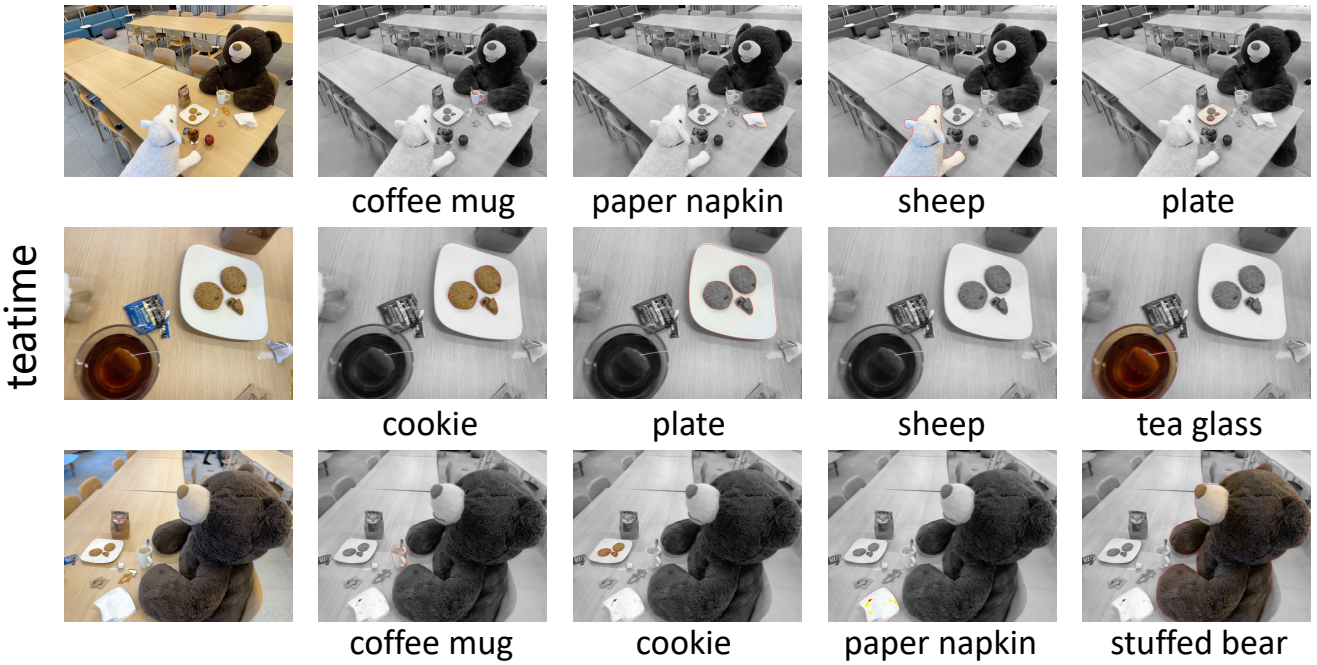

Figure 7. 3D segmentation in the *teatime* scene.

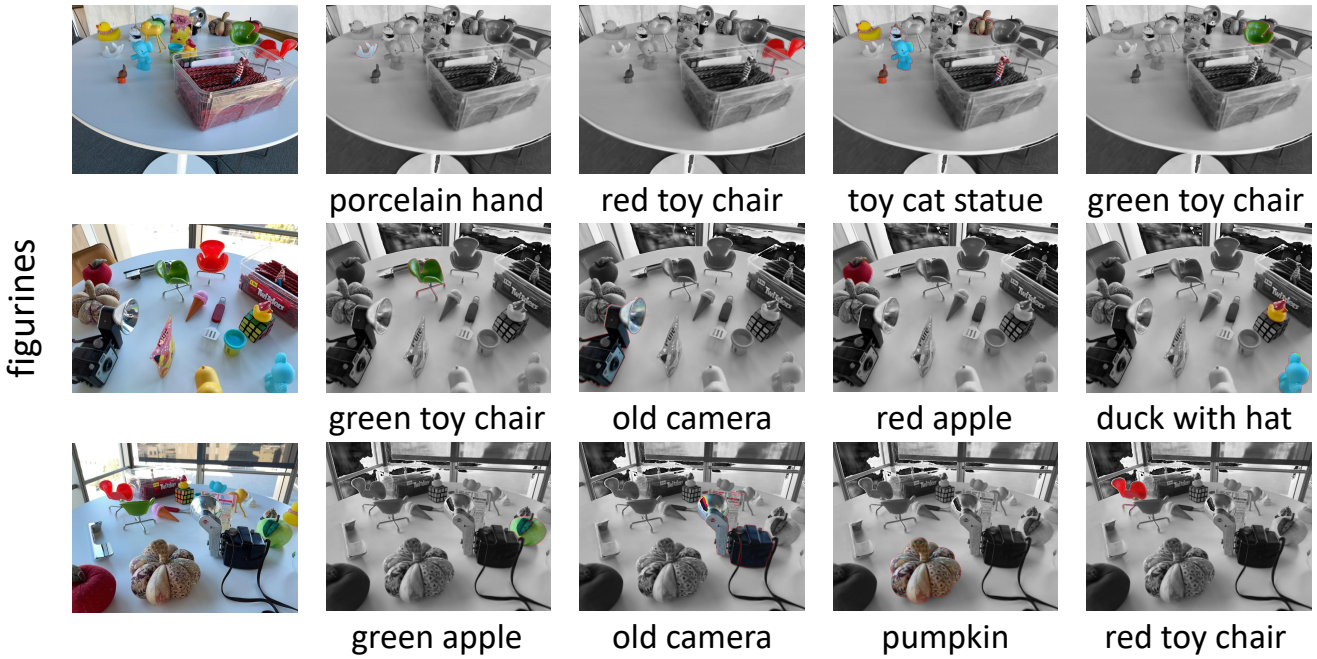

Figure 8. 3D segmentation in the *figurines* scene.