Online Language Splatting

Supplementary Material

7. Super-Resolution Decoder Architecture

We illustrate the detailed network architecture of our super-resolution decoder, as shown in Fig. 8. Since the design only includes low-cost CNN layers, this module achieves real-time. Together with the pixel-wise CLIP encoder, they compose the real-time high-resolution CLIP embedding module adopted in our online language splatting framework. For training, we use a combination of losses to ensure high-resolution feature quality and semantic coherence. The loss function is defined as:

$$\mathcal{L} = 0.8 \cdot \mathcal{L}_{\text{cosine}} + \mathcal{L}_{\text{L1}} + 0.01 \cdot \mathcal{L}_{\text{TV}},$$

- $\mathcal{L}_{\text{L1}}$ is the L1 loss, computed as:

$$\mathcal{L}_{\text{L1}} = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{y}_{\text{pred}} - \mathbf{y}_{\text{gt}}|,$$

- $\mathcal{L}_{\text{cosine}}$ is the cosine similarity loss, computed as:

$$\mathcal{L}_{\text{cosine}} = 1 - \frac{\mathbf{y}_{\text{pred}} \cdot \mathbf{y}_{\text{gt}}}{\|\mathbf{y}_{\text{pred}}\| \|\mathbf{y}_{\text{gt}}\|},$$

- $\mathcal{L}_{\text{TV}}$ is the total variation loss, computed as:

$$\mathcal{L}_{\text{TV}} = \sum_{i,j} |\mathbf{y}_{\text{pred}}(i,j) - \mathbf{y}_{\text{pred}}(i,j+1)|$$

$$+ \sum_{i,j} |\mathbf{y}_{\text{pred}}(i,j) - \mathbf{y}_{\text{pred}}(i+1,j)|.$$

where $\mathbf{y}_{\text{pred}}$ is the predicted high-resolution feature map, and $\mathbf{y}_{\text{gt}}$ is the ground-truth feature map. The $\mathcal{L}_{\text{TV}}$ loss penalizes spatial discontinuities to ensure smoothness [8].

SRD is supervised using labels that are created from SAM-generated masks. For each image, multiple points are sampled, clustered, and refined to produce the most accurate mask, which is then propagated consistently across the labels. We train the model on the COCO and Omni datasets, leveraging hierarchical features and supervised labels to help the network learn to associate boundaries and propagate information effectively. This enables the network to produce high-quality, generalizable language features.

8. Comparison to Language-GS SoTA methods on Replica Per Scene

We present the complete comparison to Language-GS SoTA methods on each scene of Replica Dataset in Table 7. As observed, our method achieve overall SoTA in both mIOU and

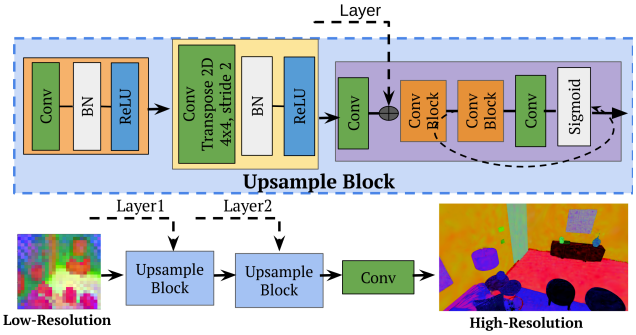

Figure 8. **Super-Resolution Decoder (SRD) Architecture.** The architecture consists of multiple layers designed to transform low-resolution input features into high-resolution outputs. The process begins with Upsample Blocks, each composed of convolutional layers, batch normalization (BN), ReLU activation, and ConvTranspose layers for spatial upscaling. After successive upsampling with the fusion of encoder intermediate layers' outputs, the output passes through a final convolutional block and sigmoid activation to produce the high-resolution feature map. This decoder refines low-resolution language features into detailed, pixel-aligned high-resolution maps for enhanced spatial understanding.

Table 6. **Top 10 Labels Used for Evaluation in Each Scene of the Replica Dataset.**

Scene	Top 10 Labels
Office0	wall, rug, table, blinds, sofa, tv-screen, chair, floor, door, bin
Office1	wall, floor, pillow, blanket, blinds, desk, desk-organizer, monitor, table, chair
Office2	wall, floor, table, sofa, panel, cushion, chair, tv-screen, bottle, tissue-paper
Office3	floor, table, wall, window, chair, sofa, tablet, cushion, door, switch
Office4	wall, floor, chair, ceiling, window, bench, panel, tv-screen, table, clock
Room0	wall, window, floor, sofa, cushion, table, rug, lamp, book, indoor-plant
Room1	wall, window, blinds, floor, blanket, lamp, ceiling, comforter, night-stand, picture
Room2	wall, chair, floor, plate, vase, window, table, indoor-plant, rug, shelf

LOC cross all scenes. On the ohter hand, certain inconsistency is also observed cross views. This may stem from varying domain gaps between testing scene and AE pretraining domains. The online AE, pretrained on the COCO and fine-tuned online, exhibits consistent results. In contrast, rows 1 and 2 (ours w/o online) use AEs trained on other Replica scenes, where greater divergence from the testing scene may cause inconsistency.

For evaluation, we utilize the following top 10 labels per scene:

Table 7. **Comprehensive evaluation on language mapping quality across Replica scenes**. Our method is evaluated against offline SoTA Lang-GS methods on the Replica dataset. We also analyze the impact of our key modules: Super-Resolution Decoder (SRD) and Online Learned AutoEncoder (OLAE) in CLIP Compression. Specifically, for versions requiring in-domain fine-tuning, two scenes from each column are held as testing scenes, while the remaining scenes are used for training. best , second-best]

Method	Modules		Room0		Room1		Room2		Office0		Time
	SRD	OLAE	mIOU	Loc	mIOU	Loc	mIOU	Loc	mIOU	Loc	
LangSplat [36]	–	–	0.356	0.710	0.459	0.779	0.381	0.641	0.433	0.763	2.8 m/fr
Feature3DGS [60]	–	–	0.487	0.677	0.301	0.812	0.353	0.800	0.342	0.661	2.3 m/fr
LEGaussian [42]	–	–	0.346	0.801	0.259	0.544	0.270	0.662	0.082	0.651	32.1 s/fr
Ours	✗	✗	0.320	0.716	0.498	0.838	0.405	0.760	0.397	0.761	0.8 s/fr
	COCO	✗	0.405	0.788	0.554	0.850	0.497	0.832	0.457	0.805	
	COCO	✓	0.389	0.773	0.493	0.832	0.576	0.833	0.454	0.758	
	Omni	✗	0.414	0.706	0.499	0.876	0.534	0.860	0.405	0.737	
	Omni	✓	0.552	0.810	0.505	0.939	0.493	0.824	0.433	0.774	

Method	Modules		Office1		Office2		Office3		Office4		Time
	SRD	OLAE	mIOU	Loc	mIOU	Loc	mIOU	Loc	mIOU	Loc	
LangSplat [36]	–	–	0.345	0.648	0.483	0.805	0.482	0.754	0.401	0.661	2.8 m/fr
Feature3DGS [60]	–	–	0.254	0.495	0.387	0.863	0.337	0.879	0.414	0.854	2.3 m/fr
LEGaussian [42]	–	–	0.354	0.414	0.178	0.680	0.267	0.943	0.204	0.766	32.1 s/fr
Ours	✗	✗	0.219	0.502	0.450	0.830	0.481	0.838	0.431	0.790	0.8 s/fr
	COCO	✗	0.272	0.393	0.570	0.847	0.553	0.919	0.492	0.824	
	COCO	✓	0.357	0.525	0.574	0.820	0.495	0.766	0.498	0.765	
	Omni	✗	0.388	0.674	0.610	0.889	0.455	0.802	0.455	0.802	
	Omni	✓	0.357	0.734	0.522	0.826	0.578	0.887	0.458	0.812	

9. Comparison to SLAM-GS SoTA methods on Replica Per Scene

We present the complete SLAM-GS evaluation results for each scene in Table 8. As observed, although our method incorporates additional open-vocabulary language mapping, it maintains the novel view rendering quality of the baseline MonoGS. Overall, our approach achieves state-of-the-art (SoTA) performance in PSNR and LPIPS metrics.

10. Open-Vocabulary Evaluation

We compare our method with other offline Lang-GS methods using GPT-generated labels on Replica. We use GPT-4o with the following prompt: "Describe the image with 5 vocabularies for each image to test object segmentation." We randomly select 30 images from 8 Replica sequences and generate labels, which are then used as prompts to query objects for these images. To generate segmentation masks and ground truth, we use Grounded SAM [38], leveraging the GPT-selected open-vocabulary (OV) labels as queries.

The evaluation results, comparing our method against offline SoTA Lang-GS methods, are presented in Table 10. While our method ranks secondary to LangSplat, it still outperforms Feature3DGS and LEGaussian by large margins and operates over $40\times$ faster, highlighting its efficiency and strong open-vocabulary segmentation performance.

We found that the key factors determining generalization capability to open-vocabulary (OV) objects is the resolution of feature maps used for GS mapping. As observed, the GPT-generated labels include many tiny objects such as "thermostat", "wall outlet", and "digital clock", which are difficult to detect in low-resolution feature maps (See Fig. 10 for examples.) Our method operates at a spatial resolution (192×192) using SRD, which is much higher compared to the pixel-wise encoder output (32x32), but remains constrained by the speed requirement for online integration of language features into 3DGS. This resolution may pose challenges for detecting tiny objects, however it provides a significant advantage in running speed, making our approach suitable for online SLAM applications. In contrast, LangSplat operates at full-resolution feature maps (1200×680), embedding them directly into 3DGS, which enhances tiny object detection but comes at the cost of a much slower runtime, making it unsuitable for real-time SLAM applications.

We acknowledge tiny object detection as a limitation of our current approach and discuss it further as part of our future work in Sec. 15.

Additionally, we evaluate the open-vocabulary segmentation of our model to determine whether it preserves the ability to segment objects using novel textual descriptions as prompts beyond the original COCO vocabulary. To test this, we randomly sample 100 COCO test images and use ChatGPT (GPT-4o) to generate semantically richer de-

Table 8. **Per Scene Evaluation of SLAM-3DGS on Replica**. Our method is evaluated against other SLAM-3DGS approaches based on novel view rendering quality and camera localization error (ATE in cm). [Key: best , second-best]

Method	w/ Lang.	Room0				Room1				Room2				Office0			
		PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓
SplaTAM [15]	✗	32.31	0.974	0.072	0.47	33.36	0.966	0.101	0.42	34.78	0.983	0.073	0.32	38.16	0.982	0.084	0.46
RTG-SLAM [35]	✗	31.56	0.967	0.131	0.20	34.21	0.979	0.105	0.18	35.57	0.981	0.115	0.13	39.11	0.990	0.068	0.22
MonoGS [27]	✗	33.36	0.941	0.086	0.458	33.58	0.942	0.086	0.424	34.12	0.950	0.081	0.490	40.91	0.980	0.045	0.615
Ours	✓	33.38	0.940	0.085	0.325	33.46	0.941	0.079	0.416	34.35	0.952	0.075	0.483	40.91	0.978	0.048	0.550

Method	w/ Lang.	Office1				Office2				Office3				Office4			
		PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓
SplaTAM [15]	✗	38.49	0.980	0.095	0.24	31.66	0.962	0.102	0.28	29.24	0.948	0.123	0.39	31.54	0.946	0.157	0.56
RTG-SLAM [35]	✗	40.24	0.992	0.075	0.12	33.54	0.981	0.128	0.22	36.48	0.984	0.117	0.20	35.43	0.982	0.109	0.19
MonoGS [27]	✗	39.77	0.976	0.049	0.327	33.81	0.907	0.114	0.341	35.17	0.954	0.058	0.303	35.02	0.952	0.082	0.405
Ours	✓	39.60	0.976	0.044	0.382	33.05	0.901	0.125	0.396	34.98	0.955	0.053	0.203	36.75	0.957	0.063	0.423

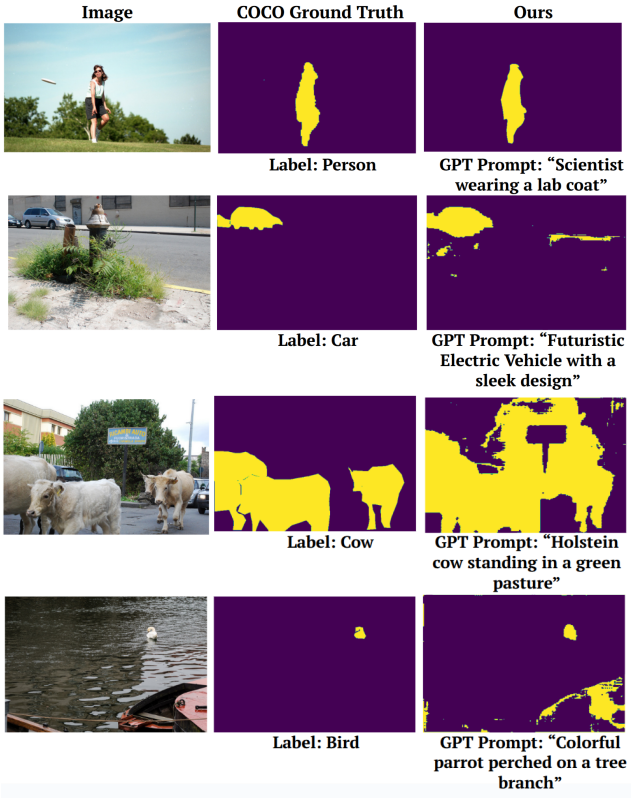

Figure 9. Open-vocabulary segmentation. **Left**: COCO ground truth segmentation. **Right**: Segmentation output of our module using GPT-generated novel vocabulary prompt.

scriptions for each label, such as replacing "car" with "futuristic electric vehicle with a sleek design". Using our trained model, which leverages CLIP-based feature representations, we generate segmentation masks for both the COCO labels and the GPT-generated descriptions. The model achieves an mIoU of 0.389 with COCO labels and 0.392 with GPT labels (Table 9, demonstrating that it maintains segmentation performance regardless of textual variation. The qualitative results (Fig. 9 compare the COCO ground-truth segmentation with our model's segmentation using GPT-generated novel descriptions. The results support the ability to generalize beyond COCO labels. These findings confirm that despite being trained on COCO for upsampling, the model effectively operates in an open-vocabulary setting.

Table 9. Comparison of mIoU performance using COCO dataset labels and ChatGPT-generated novel vocabulary on 100 randomly sampled test images. The similar mIoU scores indicate that our method preserves CLIP's open-vocabulary capabilities.

Method	mIOU
COCO labels	0.389
GPT novel labels	0.392

11. More Visualizations

In this section, we provide additional visualization results in Figs. 12- 18. To ensure a fair comparison, we increase LangSplat's code size from 3 to 15 and upgrade OpenCLIP's [12] feature dimension from 512 to 768.

Details of heat map results and evaluation metrics. We display 2D heat maps as query results throughout this work. For each text query, LangSplat generates three Gaussian relevancy language features, while our method produces pixel-level language features through our high-resolution model. To calculate localization and IoU metrics and reduce the impact of outliers, similar to LangSplat, we apply a mean convolution filter with a kernel size of 20 to smooth the values in the language feature maps. The final score is determined by selecting the maximum relevancy score.

We used a score threshold of 0.4 for LangSplat and 0.5 for our method. We tuned the threshold that shows the best

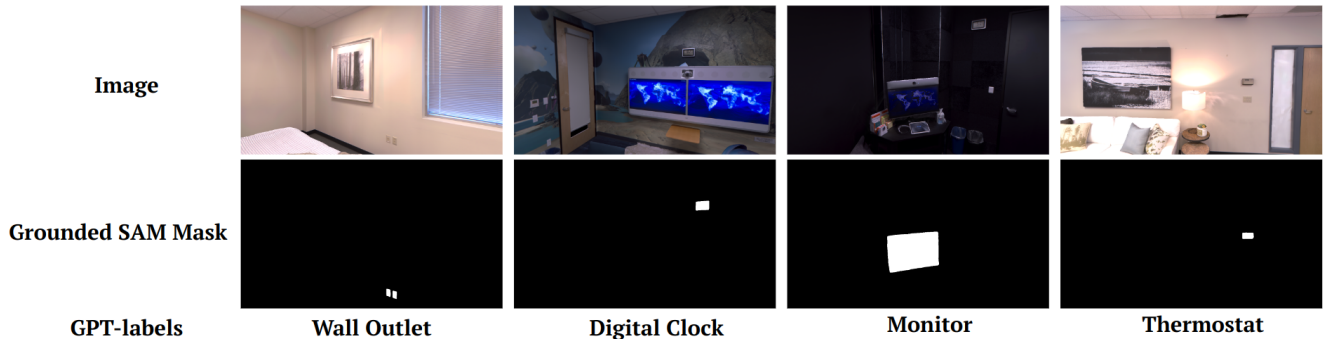

Figure 10. Examples of GPT-generated object labels and masks from Grounded SAM.

Table 10. **Comparison on GPT-generated labels for Replica**. [Key: best, second-best]

Method	GPT-labels		FeatureMap Res.	Time
	mIOU	Loc		
LangSplat [36]	0.660	0.880	1200×680	2.8 min/fr
Feature3DGS [60]	0.489	0.600	480×360	2.3 min/fr
LEGaussian [42]	0.241	0.703	184×110	32 s/fr
Ours	0.539	0.765	192×192	0.8 s/fr

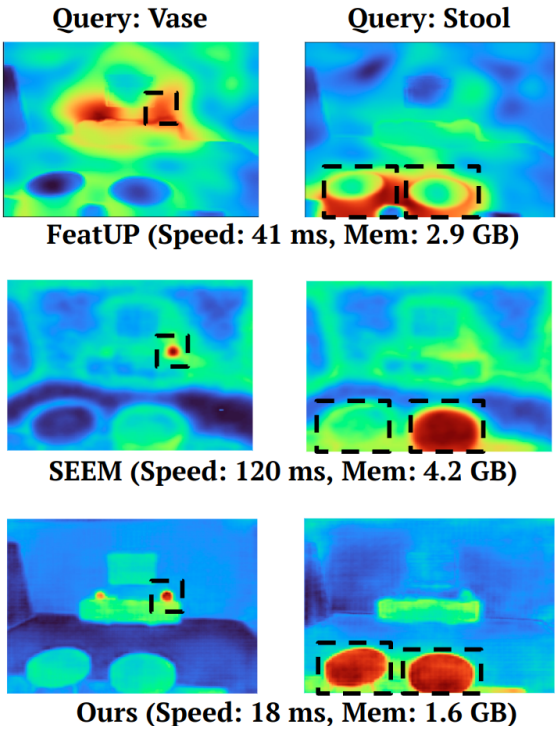

Figure 11. We provide a visual comparison for feature maps by FeatUP [8], the open-vocabulary segmentor from SEEM [62], and our HR module. As observed, despite its simplicity, our HR module achieves the highest feature quality at the lowest cost and fastest speed. We believe our simple and highly effective design provides valuable new insights. Black box: groundtruth.

object boundaries for each method for a fair comparison. Values below the threshold are classified as background, and those above it generate binary maps.

Table 11. Comparison of various language code sizes on Replica Office0 (Of0) and Room2 (Rm2) sequences.

Size		128	64	32	20
Rm2	MSE	0.0064	0.0065	0.0065	0.0068
	MAE	0.0497	0.0502	0.0505	0.0620
	Cosine Sim.	0.9817	0.9764	0.9690	0.9000
Of0	MSE	0.0059	0.0060	0.0068	0.0074
	MAE	0.0439	0.0445	0.0484	0.0570
	Cosine Sim.	0.9823	0.9750	0.8790	0.7600

Table 12. Comparison of IOU and Localization (Loc) accuracy across different code sizes (15, 6, 3) for Replica Office3 (Of3) and Office4 (Of4) sequences.

Code Size	Of3 IOU	Of3 Loc	Of4 IOU	Of4 Loc
15	0.495	0.766	0.498	0.765
6	0.485	0.708	0.490	0.701
3	0.480	0.690	0.487	0.693

12. Language Compression

12.1. Study on Code Size

We study the generalizability of autoencoder code sizes trained on COCO and tested on Replica sequences (*Room2 (Rm2)* and *Office0 (Of0)*) (see Table 11), evaluating reconstruction quality using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Cosine Similarity. MSE measures the squared differences between original and reconstructed features, MAE quantifies the absolute deviation, and Cosine Similarity assesses the alignment of feature vectors. As the code size decreases, we observe an increase in errors, which is expected due to the trade-off between compactness and information retention. Smaller code sizes cre-

Table 13. **Comparison on 3D localization evaluation**. We counts a query as failure when a distance is larger than the CD/EMD's population mean plus 2× population standard deviation. Failures are excluded from the average and reported separately. Of: Office; Rm: Room from the Replica Dataset.

Average CD	Of0		Of1		Of2		Of3		Of4		Rm0		Rm1		Rm2		Overall	
	CD	Failure	CD	Failure	CD	Failure	CD	Failure	CD	Failure	CD	Failure	CD	Failure	CD	Failure	CD	Total count
LangSplat [36]	1.175	1	0.764	3	1.232	1	0.828	1	1.450	1	0.942	1	1.044	1	0.342	1	0.972	**10**
Ours	0.620	1	0.922	3	0.804	1	0.284	1	1.380	0	0.657	1	0.582	2	0.567	1	**0.736**	**10**

Average EMD	Of0		Of1		Of2		Of3		Of4		Rm0		Rm1		Rm2		Overall	
	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Failure	EMD	Total count
LangSplat [36]	7.369	2	4.745	4	2.949	1	5.512	1	17.42	2	4.549	1	3.824	2	4.109	2	6.310	15
Ours	1.574	1	2.292	5	9.001	1	7.949	1	9.157	1	1.498	1	13.987	1	0.100	3	**5.695**	**14**

Table 14. **Comparison of 1-stage and 2-stage methods**

Scene	**1-stage (768 → 15) Offline**	**2-stage (768 → 32 Pretrained, 32 → 15 Online)**
Room0	0.514, 0.835	0.552, 0.810
Room1	0.427, 0.839	0.505, 0.939
Room2	0.396, 0.801	0.493, 0.824
Office0	0.422, 0.761	0.433, 0.774
Office1	0.409, 0.802	0.522, 0.826
Mean	0.434, 0.808	**0.501**, **0.835**

ate more compact feature representations but often reduce structural detail and granularity in the reconstructed language feature maps, leading to losses in semantic and spatial accuracy. This happens because smaller latent spaces constrain the feature encoding, causing a loss of fine-grained variations that are essential for precise language-based localization. To balance accuracy and computational efficiency, we choose code 32, which retains sufficient semantic fidelity while remaining efficient for real-time applications.

We evaluate the impact of varying the code size of the online encoder-decoder on IOU and Localization (Loc) metrics using Replica sequences. See Table 12. To balance memory and ensure real-time feasibility, we limit the maximum code size to 15. Online training improves adaptability by fine-tuning representations for specific sequences, but its effectiveness depends on the code size. Smaller code sizes constrain the latent space, leading to a loss of fine-grained details and limiting the benefits of online training. In contrast, a code size of 15 strikes a balance between compression and capacity, allowing the model to leverage online adaptability while preserving semantic and spatial accuracy.

12.2. Online Compression

Table 14 presents a comparison between single-stage and two-stage compression methods across multiple scenes.

Table 15. **Effects of disentangling GS parameters into color and language modes.**

Disentangled Mode			**Image Rendering**		
Separate α	Separate $\boldsymbol{R}$	Separate $\boldsymbol{S}$	PSNR↑	SSIM↑	LPIPS↓
✗	✗	✗	31.23	0.901	0.197
✗	✓	✓	31.79	0.915	0.177
✓	✗	✗	31.75	0.918	0.180
✓	✓	✗	32.80	0.929	0.146
✓	✗	✓	33.57	0.939	0.118
✓	✓	✓	**35.89**	**0.957**	**0.060**

The results indicate that the two-stage method generally offers improvements in both mIOU and localization accuracy, with certain scenes like Room1 demonstrating more noticeable gains (mIOU from 0.427 to 0.505 and localization accuracy from 0.839 to 0.939). While the single-stage method is simpler and more memory-efficient, it may lose important language features due to aggressive compression (768D directly to 15D). Conversely, the two-stage method introduces an intermediate compression stage, preserving these language features but at the cost of increased complexity. Therefore, the choice between these two methods ultimately depends on the user's specific requirements and constraints regarding accuracy versus resource utilization.

13. Detailed Study on Disentanglement GS Parameters

First, as a more detailed version of the 3D localization evaluation compared to Table 5, we present per-class evaluation results in Table 16. As observed, the disentangled optimization leads to better overall performance and significant improvements on some classes.

Next, we study different strategies in GS parameter disentanglement, including separating α, $\boldsymbol{R}$ and $\boldsymbol{S}$ into color and language modes. In the forward rendering, we splat 3D Gaussians onto the 2D space and conduct alpha compositions using respective mode parameters to render color and languages.

In the back-propagation, if α is disentangled into color (c) and language (f) modes, we then calculate gradients by

$$\frac{\partial \mathcal{L}}{\partial \alpha_i^c} = \frac{\partial \mathcal{L}}{\partial C}\frac{\partial C}{\partial \alpha_i^c} + \frac{\partial \mathcal{L}}{\partial D}\frac{\partial D}{\partial \alpha_i^c}, \quad \frac{\partial \mathcal{L}}{\partial \alpha_i^f} = \frac{\partial \mathcal{L}}{\partial F}\frac{\partial F}{\partial \alpha_i^f}, \tag{8}$$

where $\frac{\partial \mathcal{L}}{\partial \alpha_i^c}$ and $\frac{\partial \mathcal{L}}{\partial \alpha_i^f}$ further propagates to $\boldsymbol{R}_i$ and $\boldsymbol{S}_i$ via the world-coordinate 3D covariance matrix $\boldsymbol{\Sigma}_i^{c/f}$. If α is not disentangled, the gradient terms in Eq. (8) are added together.

The same rule applies to $\boldsymbol{R}_i$ and $\boldsymbol{S}_i$. If it is disentangled, its color or language mode's gradients are separately computed from (added or separated) $\frac{\partial \mathcal{L}}{\partial \alpha_i}$. If not, its gradients are added by both color and language modes.

Results shown in Table 15 validates that disentangling $\boldsymbol{R}$, $\boldsymbol{S}$, and α works the best to preserve the highest image quality. Comparing the fourth and fifth rows, one can also find disentangling $\{\alpha, \boldsymbol{S}\}$ has better effects than $\{\alpha, \boldsymbol{R}\}$. This echoes the observation in the main paper Fig. 4: language mode prefers larger scales to cover more areas that belong to the same language codes, compared with color rendering that needs smaller Guassians to represent finer textures. Thus, disentangling $\boldsymbol{S}$ shows better performances. As a further discussion, disentangling the world-coordinate 3D mean $\boldsymbol{\mu}$ will produce much more Gaussians that attempt to fit in color and language views separately. The setting consumes 68% more memory than disentangling $\boldsymbol{R}$, $\boldsymbol{S}$, and α and cannot finish training on a Replica sequence on a RTX-3090 GPU.

14. More 3D Localization Evaluation

Following the 3D localization experiment in the main paper, we show more 3D localization evaluation on the 8 sequences of the Replica Dataset. We adopt top-10 frequent categories in each sequence including objects and area, counted by visible pixels. Quantitative results are provided in Table 13. Note that language label ambiguity exists in Replica's annotations. For instance, in the "tv-screen" example in Fig. 20, the groundtruth only counts in the border areas and leaves out the center display areas as "undefined". Another example is in Fig. 19 the ceiling areas exclude the lights, which is also ambiguous in defining the ceiling regions. This can result in significantly larger point cloud distances, as measured by the CD and EMD metrics, when queries return objects that do not align with the definitions of the annotation system. To alleviate this, for CD and EMD, we set a threshold that directly counts a query as failure, when a distance exceeds the metric's population mean plus $2\times$ population standard deviation. The failures are excluded from the average and reported separately as counts of failure. Results are shown in Table 13. Our method performs better than LangSplat on overall CD/EMD with equal or smaller failure counts.

In Fig. 19 and 20, we visualize more 3D localization results by language queries, which extend Fig. 6 in the main paper with the same TSDF procedure to reconstruct meshes.

15. Limitations and Future Work

This work focuses on static scenes, which may limit its applicability to dynamic environments where objects or spatial configurations change over time. Additionally, both our method and LangSplat are susceptible to false positives for objects that are visually or semantically similar. We notice, for smaller objects SAM generated masks tends to produce more crisp results, whereas for room-sized objects our method shows better localization accuracy due to globally trained pixel-wise CLIP embedding.

In future work, we aim to extend our approach to dynamic scenes by incorporating mechanisms to handle temporal changes and object motion. Additionally, we plan to explore uncertainty quantification for language features to better evaluate and communicate the reliability of predictions. This improvement would enhance practical use cases, such as robotic navigation and interaction, where confidence in localization is critical.

Table 16. **Effects of Disentangled GS Parameters (Category-Wise)**. Here, we show category-wise results for language-queried 3D localization on the Replica Room-0 subset.

Method	**3D Language Query**															
	cabinet		cushion		stool		rug		lamp		wall		ceiling		**Average**	
	CD ↓	EMD↓	CD ↓	EMD↓	CD↓	EMD↓	CD↓	EMD↓	CD↓	EMD↓	CD↓	EMD↓	CD↓	EMD↓	CD↓	EMD↓
Joint RGB-L	0.042	**0.019**	1.252	3.022	**0.128**	**0.007**	0.283	**0.866**	**0.579**	9.517	0.110	0.118	0.295	**0.033**	0.384	1.940
Disentangled	**0.040**	0.053	**0.554**	**0.032**	0.196	0.399	**0.256**	1.494	1.222	**4.083**	**0.087**	**0.050**	**0.269**	0.707	**0.375**	**0.974**

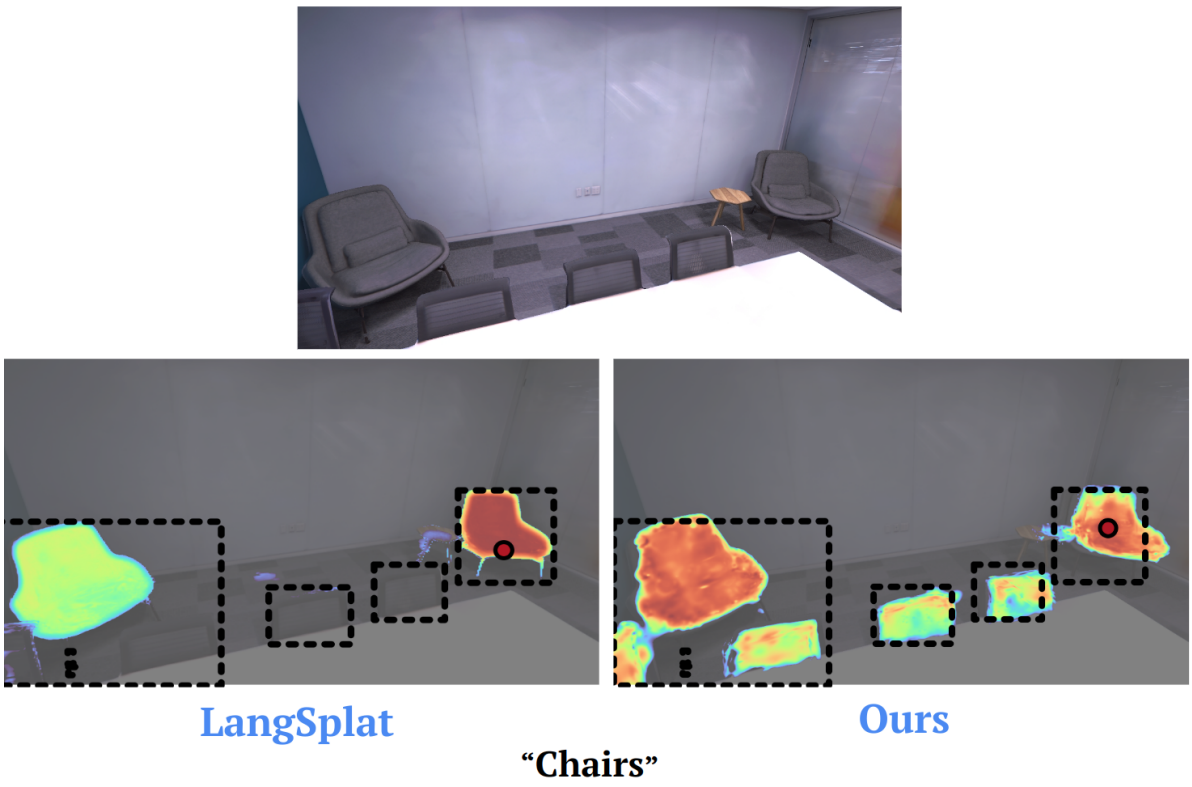

Figure 12. Our method is able to identify and segment all of the chairs, while LangSplat was only able to segment only two chairs.

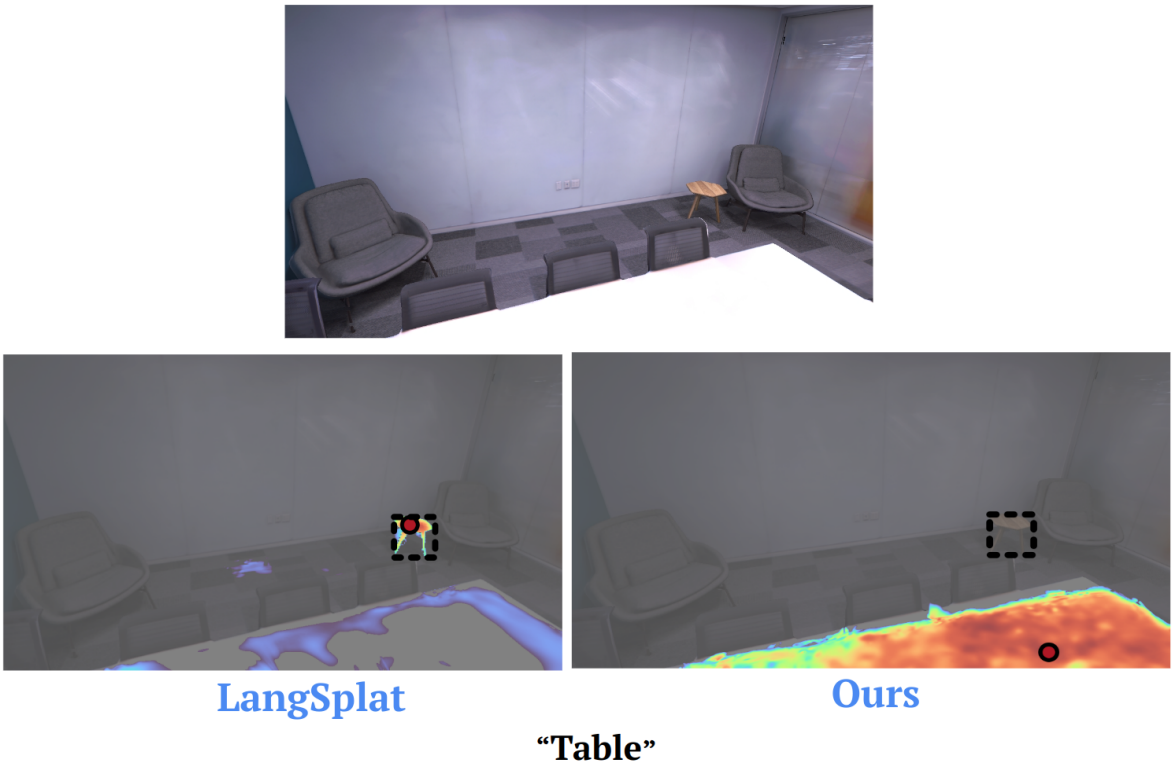

Figure 13. Comparison of segmentation results for the query ”Table”. LangSplat successfully segments the table, while our method focuses on the table top instead, leading to a segmentation error. This discrepancy is considered a failure case for our approach.

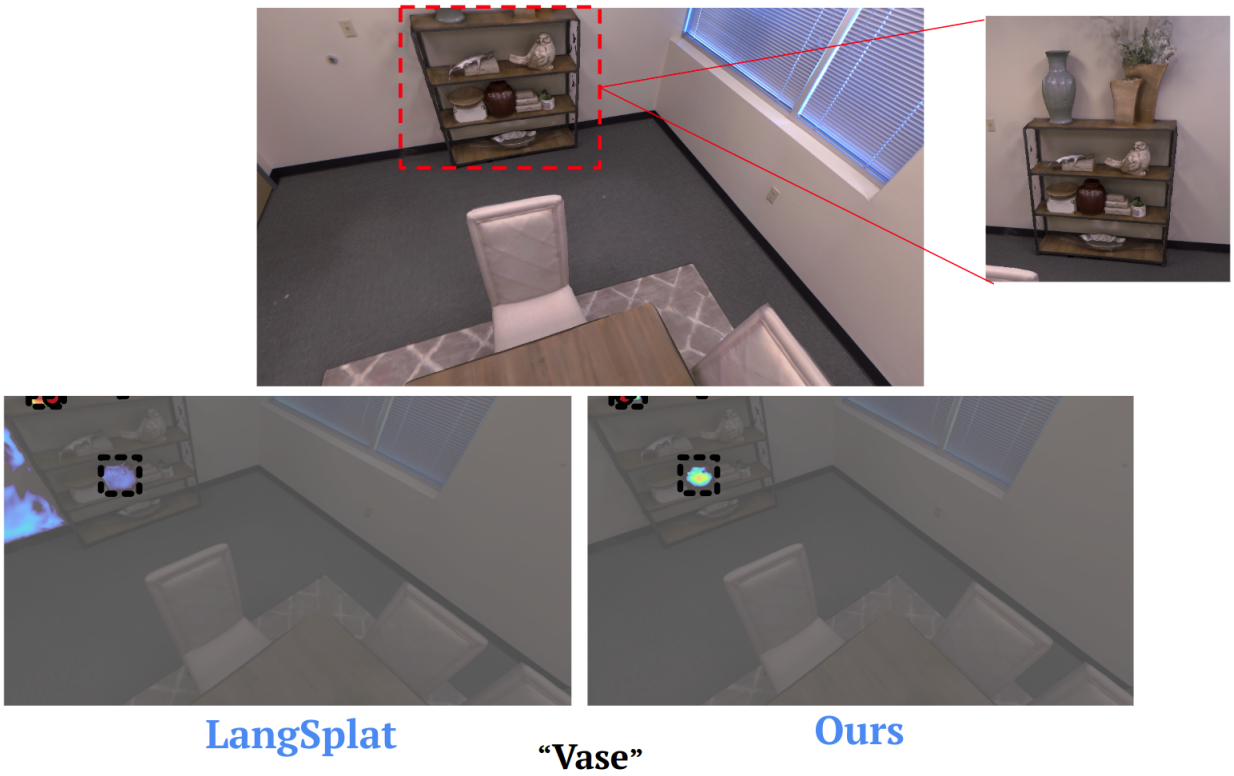

Figure 14. Both methods demonstrate 3D consistency. The top-right zoom-in image shows the vase from another frame. Our model, due to its 3D consistency, successfully detects the vase from only very limited appearance in the current frame. This highlights the ability of our method to handle occlusions or partial appearance effectively.

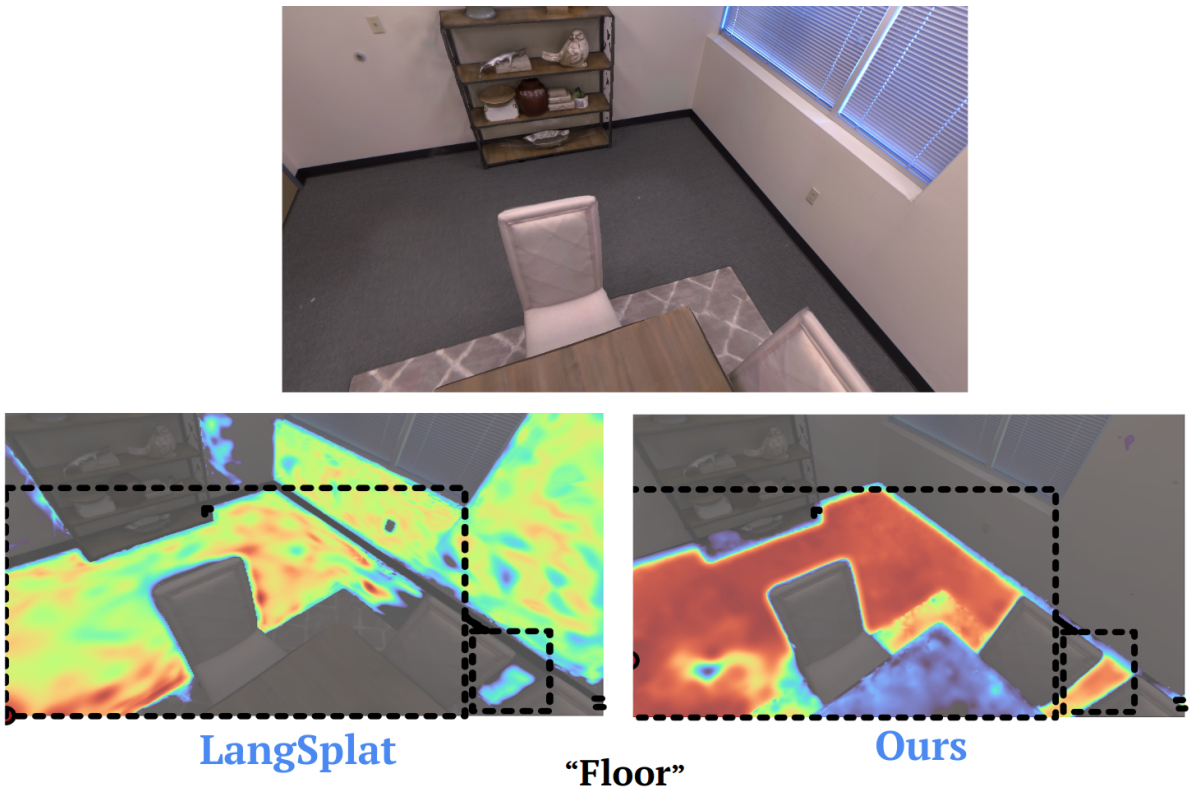

Figure 15. Comparison of floor segmentation results. LangSplat introduces outliers by incorrectly segmenting walls as part of the floor. In contrast, our method accurately segments the floor without including such outliers.

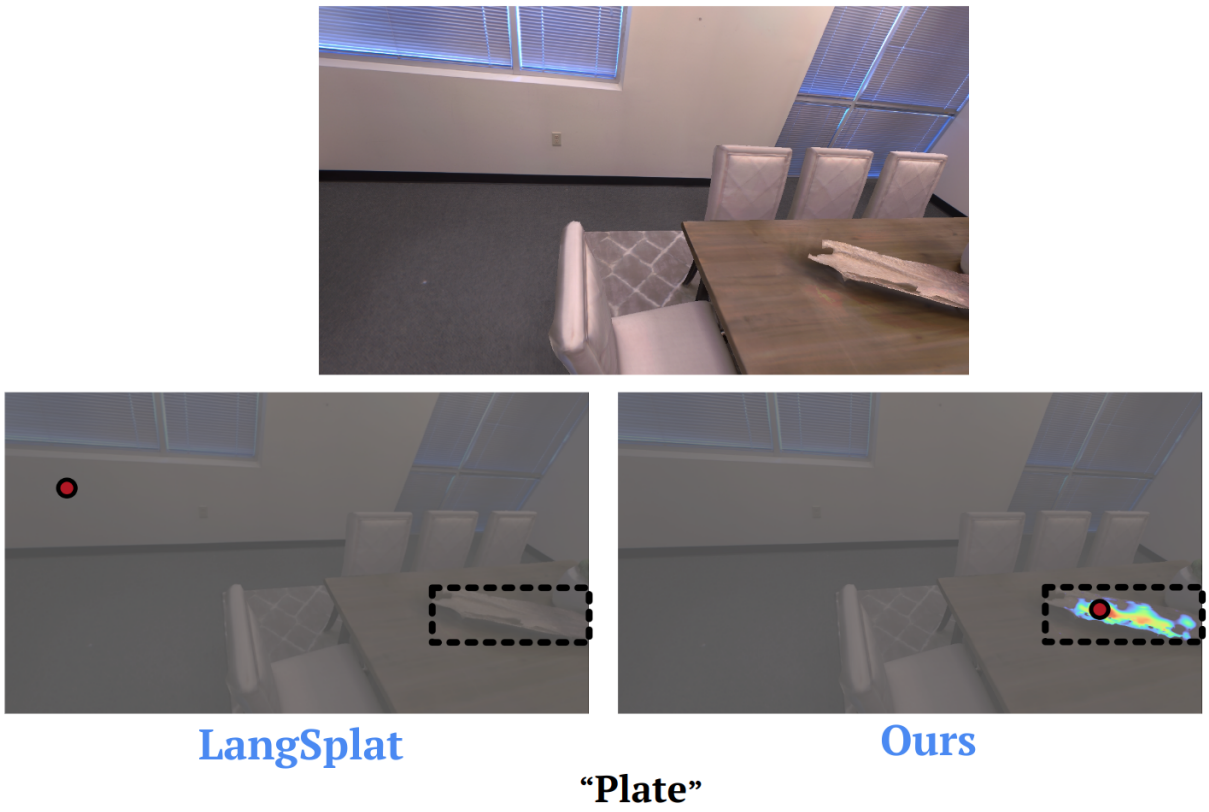

Figure 16. Comparison of plate segmentation. LangSplat fails to detect the plate, whereas our method successfully identifies and segments the plate on the table.

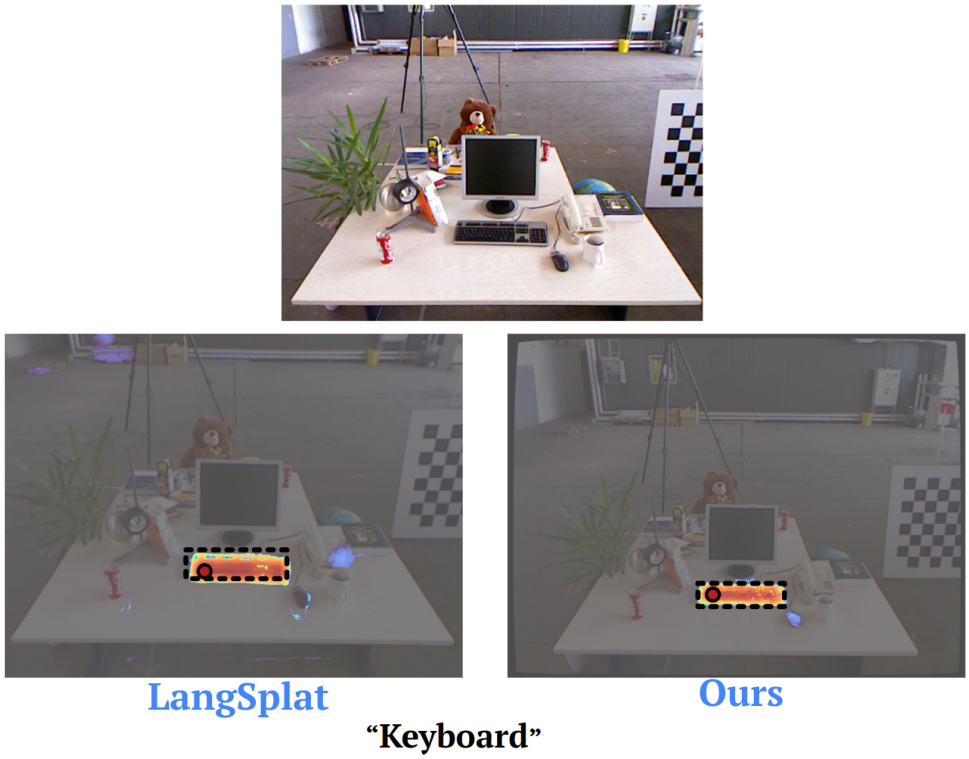

Figure 17. Comparison of ”Keyboard” query localization on TUM-RGBD.

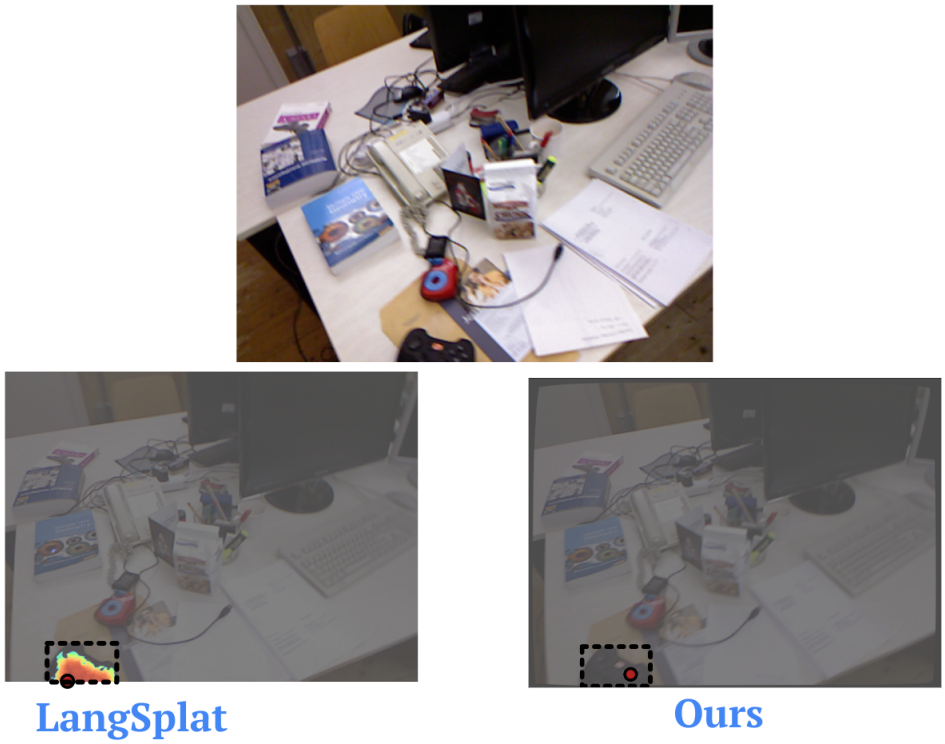

Figure 18. Comparison of ”Game Controller” query localization on TUM-RGBD.

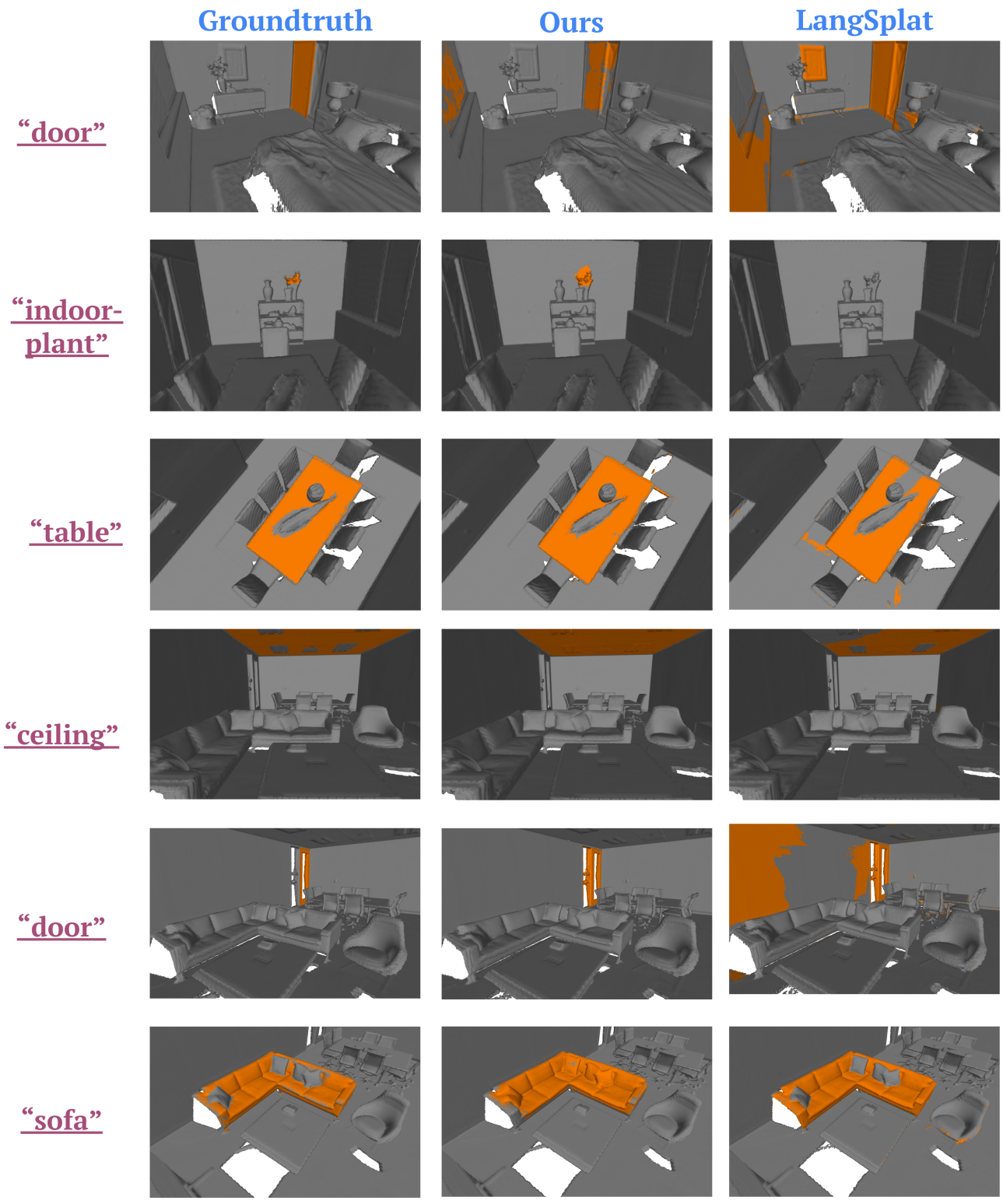

Figure 19. Comparison of 3D localization by queries on Replica sequences.

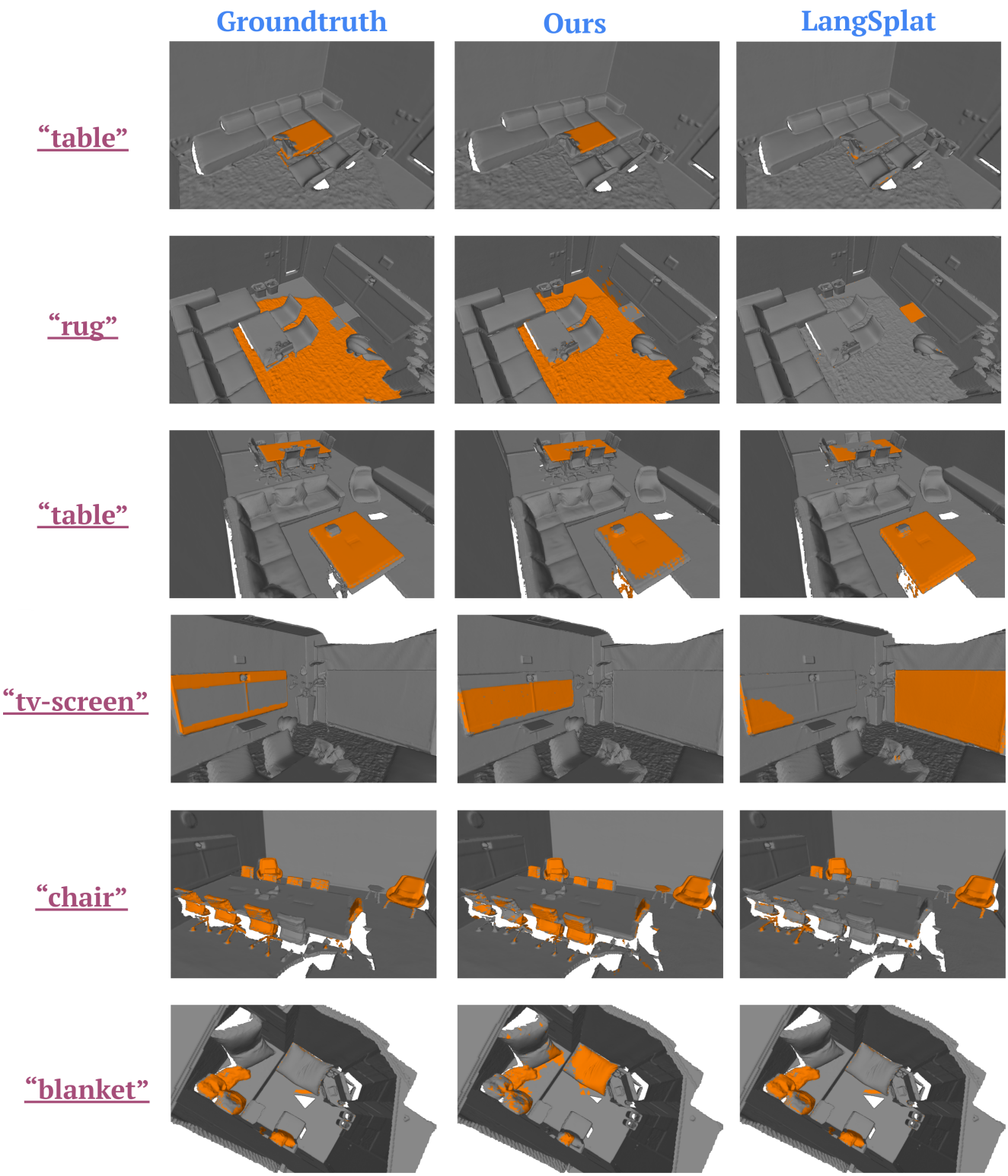

Figure 20. (Continue) Comparison of 3D localization by queries Replica sequences.