RelationField: Relate Anything in Radiance Fields

Supplementary Material

In this **supplementary material**, we first provide additional training details in Sec. A. Next, we offer further insights into our design choices for RelationField in Sec. B. Sec. C contains additional details on the scene graph extraction. We then present qualitative results for relationship segmentation and the relationship-guided 3D instance segmentation task in Sec. D. In Sec. E, we include an adaption of RelationField to Gaussian Splatting together with a quality comparison. Finally, we provide examples from our curated relationship-guided 3D instance segmentation benchmark in Sec. F.

A. Training details

To accelerate training speed, existing feature field approaches [3, 6, 7] store extracted 2D training features to disk and load them in RAM at training start for efficient retrieval at each training step instead of computing the features online. However, storing all relationship features in RAM or disk is infeasible for RelationField since for n input images of shape w, h, with m generated masks it would require storing $n \times m \times (m-1) \times w \times h \times d$ relationship features. This would result in $\sim 5.66TB$ when we store the features in FP16 and assume 10 instances per image for a scene of 200 images each with VGA resolution of 640×480. Instead, we optimize the memory resources by storing a dictionary of all relationship features in combination with a singular segmentation mask and compute the relationship map for each sampled pixel-pair using a two-step lookup in the segmentation map and then in the relationship dictionary. Using this strategy we are able to reduce the memory requirements to $\sim 500MB$ per scene when using FP16 precision. Inspired by [7], we begin training the relationship field after 2000 steps of NeRF optimization to let the geometry converge. We train for 30000 steps on a single Nvidia A100, which takes around 60 minutes and consumes around 40GB of GPU memory. The feature extraction of the object features from OpenSeg [4] and SAM [8], as well as the relationship features with GPT-4 increases the training time by about 30 minutes for the first run.

B. Design choices

Prompting. To extract textual relationships using GPT-4 [1] or Llama [2], we employ a combination of visual and textual prompting. For visual prompting, we utilize SoM [15] to overlay semi-transparent masks and numeric marks. The textual prompt consists of a two-stage approach which queries the model first to extract objects by their mark-id and then to extract relationships referenced by the previously extracted object-ids together with a relationship label. The complete prompt looks as follows:

> **1. Object Identification:** Identify all objects in the image by their tag. Create a dict that maps tag_id to class_name.
>
> **2. Affordance/Relationship Detection:** For every pair of tagged objects that are clearly related, describe the semantic relationships and affordances as a list of dictionaries using the format [s_id: #n1, subject_class: x, o_id: #n2, object_class: y, predicates: [p1, p2, ...]]. For subjects and objects sharing multiple relationships/affordances, concatenate predicates with a comma in the [predicate] field.
>
> - Avoid generic terms like "next to" for ambiguous relationships. Instead, specify relationships with precise relationships and affordances describing spatial relationships [over/under etc.], comparative relationships [larger/smaller than, similar/same type/color], functional relationships [part of/belonging to, turns on], support relationships [standing on, hanging on, lying on, attached to].
> - Do not use left/right; always use 3D consistent relationships.
> - Always combine a spatial relationship with a semantic, comparative, functional or support relationship using a comma (e.g., [A] [above, lying on] [B]).
> - For symmetrical relationships, include both directions (e.g., [A] [above] [B] and [B] [below] [A]).
> - Even for distant objects highlight if they are [same/similar/same color/same object type]
>
> **Example Output:**
> objects = {4: floor, 7: table, 12: chair, ...}
> relationships_affordances = {
> [s_id: 4, subject_class: table, o_id: 7, object_class: floor, predicates: standing on],
> [s_id: 12, subject_class: chair, o_id: 13, object_class: chair, predicates: next to, same as],
> [s_id: 6, subject_class: pillow, o_id: 8, object_class: couch, predicates: belongs to],
> ...
> }

After processing the image frames with the LLM we parse the output into a JSON format in an automatic manner.

Text encoder. To embed relationships in RelationField, we encode the output from a multi-modal LLM into the radiance field using an encoder-only language model. The choice of the encoder is important since it determines the structure and queryablity of the embedding space in the radiance field. We want an embedding space, that is highly structured and embeds similar (relationship) concepts close together, while contradictory relationships are supposed to be far apart in embedding space. In Fig. 1, we provide an analysis for different popular open-source text encoders such as CLIP [13], BERT [5], Jina-v3 [14], RoBERTa [10] and GPT-2 [12]. We have a set of 41 distinct relationships with varying semantic similarity to each other and plot their pair-wise cosine similarity in a similarity matrix. We observe that Jina-v3-

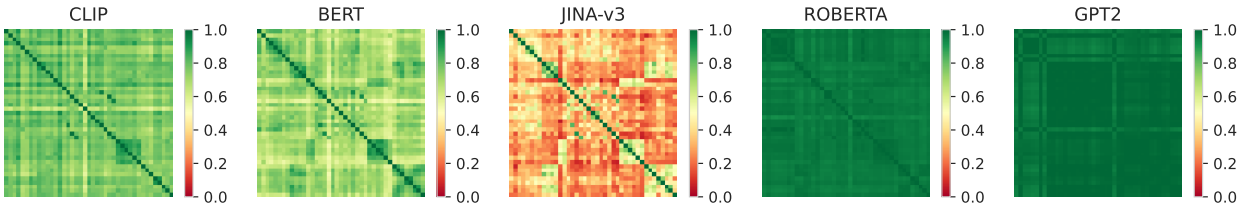

Figure 1. **Language Encoder Ablation.** We compare 5 language encoder-only model based on their separability in embedding space. For each language encoder, we plot a similarity matrix, for the pairwise cosine-similarity of 41 predicates taken from the 3DSSG dataset.

embeddings generate the most well-structured feature space, where related concepts exhibit a strong similarity, while the majority of relationships describing distinct concepts show a high degree of dissimilarity. As a counter-example, both RoBERTa and GPT2 embed all relationships in a very similar feature space, which would make fine-grained querying difficult.

Relationship direction. In Fig. 3 of the main manuscript, we present qualitative results from RelationField on 4 scenes. In these results, we present queries of the form "What is <> *standing on/attached to/similar to etc.*?". In this scenario, we are interested in the object of a *subject-predicate-object* relationship. However similarly, it can be interesting to investigate to query the subject of a *subject-predicate-object* relationship by answering the question "What is *standing on/attached to/similar to etc.* <>?" To model this question in RelationField we simply have to invert the supervision signal during training by swapping the query ray origin with the ray origin. In Fig. 2, we demonstrate different directional relationship queries for the same objects and predicate.

C. Scene Graph Construction

In Fig. 3, we visually supplement the reported process of extracting a 3D scene graph from RelationField. First, we extract groups of points from the instance field. These groups of points (Fig. 3a), serve as the queries for the RelationField and represent the subject in a *subject-predicate-object* relationship edge. In a second step, the RelationField gets evaluated on the remaining points of the point cloud given the query points and a textual relationship prompt such as lying on (Fig. 3b). The textual query represents the predicate in the *subject-predicate-object* relationship. This step returns a relationship activation map for the entire point cloud with each point having a unique relationship response. In the third step, the activations get aggregated based on the instance head (Fig. 3c). The instances that have a relationship response greater than a threshold of 0.5 represent objects in the *subject-predicate-object* relationship edge. Finally all edges for objects surpassing the threshold are added to the 3D scene graph.

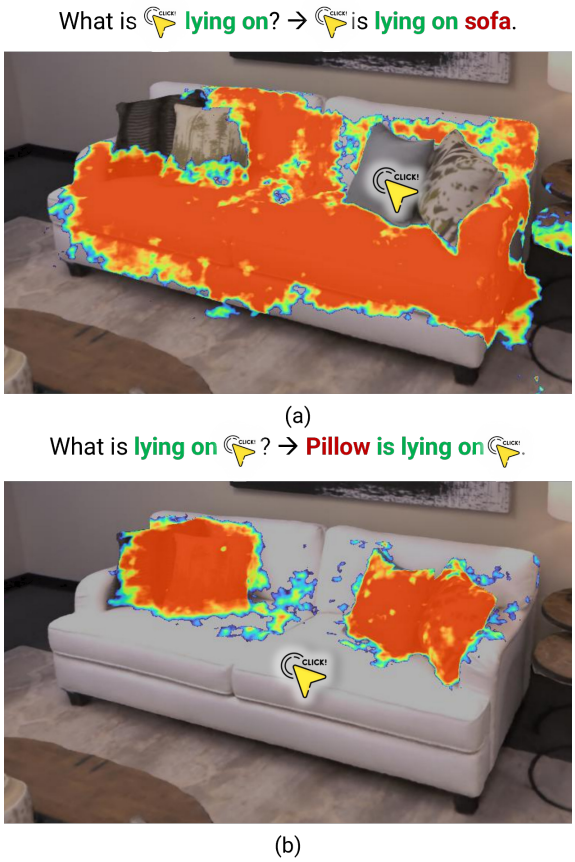

Figure 2. **Relationship Direction.** (a) visualizes the relationship response for the question "What is *standing on/lying on/similar to*?", where we localize the object in a *subject-predicate-object* relationship. While (b) visualizes the relationship response for the question "What is *standing on/lying on/similar to* ?", where we localize the subject in a *subject-predicate-object* relationship.

D. Qualitative Results

Relationship querying. In Fig. 4, we present qualitative results for 4 additional scenes for the relationship querying with RelationField.

Relationship-guided instance segmentation. In Fig. 6, we qualitatively compare the 3D instance segmentation of RelationField against OpenNeRF [3] for relationship queries

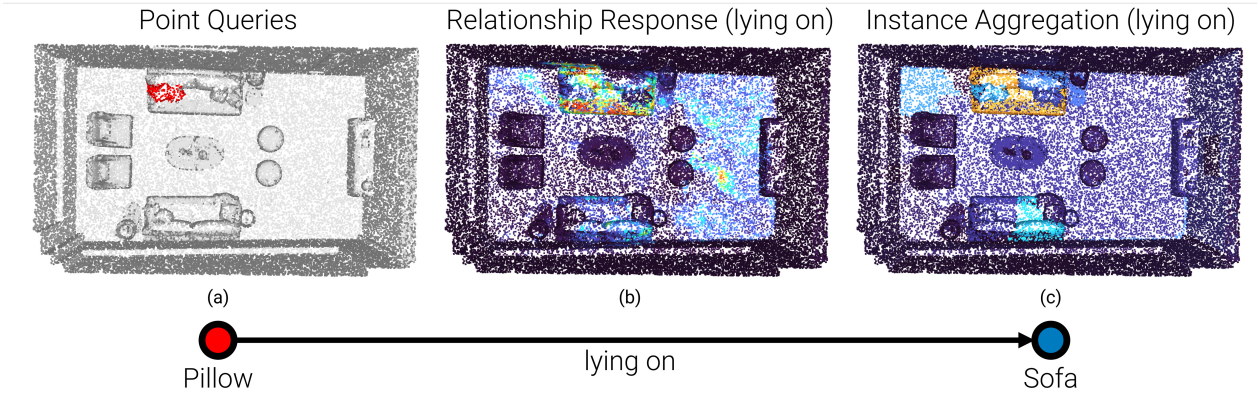

Figure 3. **Relationship Edge Construction.** To extract a 3D scene graph from RelationField, we automatically query instances (a), compute the relationship response for predicates such as "lying on" (b), and aggregate the relationship response for each instance (c). We add an edge to the scene graph for all objects whose relationship response for the subject and predicate is greater than a certain threshold.

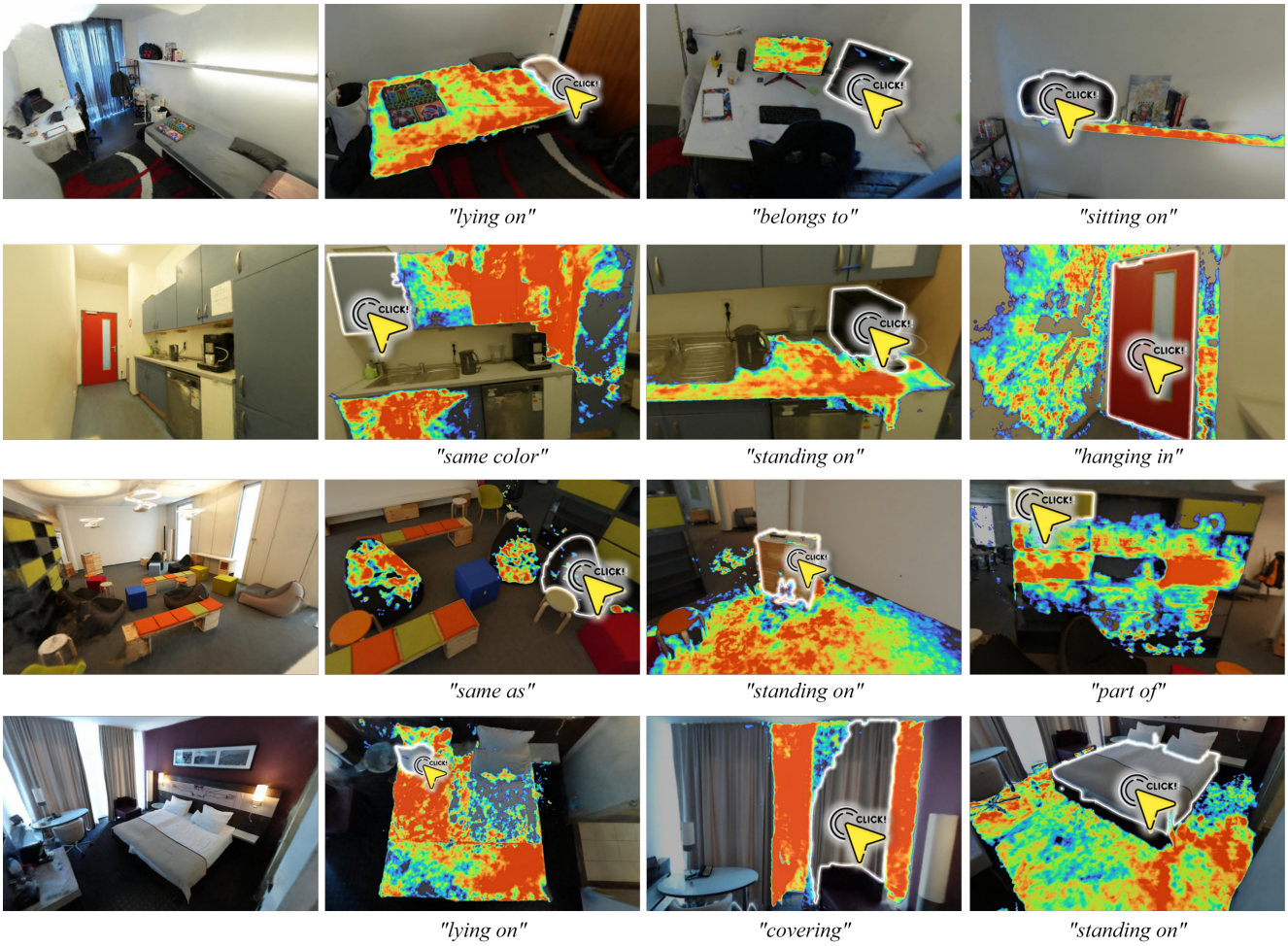

Figure 4. **Additional Qualitative Results.** We provide relationship responses for 4 additional scenes from Scannet++. The colormap visualizes the relationship response where blue is low and red is high. We visualize the relationships for the question: "What is *standing on/lying on/similar to*?"

to supplement Tab. 2. OpenNeRF produces many false positives because it gets confused with the compositional queries arising from the bag-of-words behavior of CLIP [12]. Meanwhile, RelationField uses the object information together

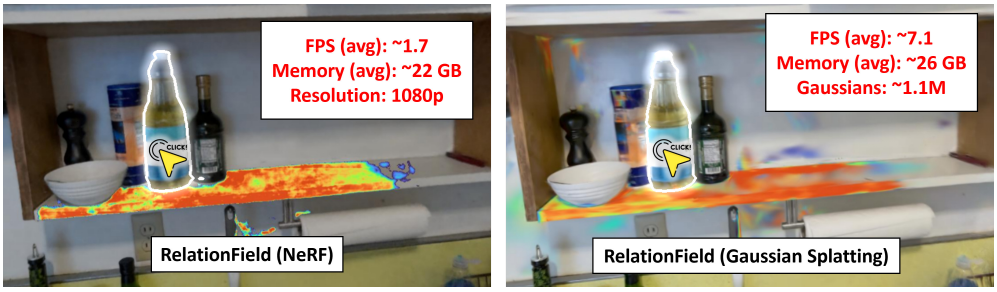

Figure 5. **RelationField w/ NeRF or w/ Gaussian Splatting geometry**. We compare the rendering speed (FPS), memory requirements and RelationField quality for the query "standing on".

with the relationship information from the prompt to accurately filter predictions that only correspond to the object in the prompt that has the described relationship.

E. Gaussian Splatting Support

We build RelationField on NeRF [11], however since our approach is independent of the underlying 3D representation, it is possible to extend RelationField to Gaussian Splatting for faster training, inference and rendering. To train RelationField, we follow [9] and initialize the Gaussian Splatting training run with the exported point cloud of the NeRF training. This results in faster convergence and fewer Gaussians leading to improved memory utilization. For RelationField with Gaussian Splatting geometry, we reformulate our relationship definition from a pair of rays to a pair of 3D Gaussian centers. In Fig. 5, we compare the rendering speed, memory requirements and RelationField quality. RelationField based on Gaussian Splatting achieves 4x faster rendering compared to its NeRF variant with a lower memory footprint. Overall, Fig. 5 shows that RelationField is independent of the underlying geometry, and both NeRF and 3DGS produce high-quality RelationFields.

F. Relationship-guided 3D Instance Segmentation Dataset

In Fig. 7, we present a subset of the annotated benchmark which we present in Sec 4.1 of the main paper. In the benchmark, we provide instance segmentations paired with textual relationship prompts. When curating the benchmark we focused on samples that appear multiple times in the scene, but which can be uniquely referenced by a relationship prompt.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[3] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. OpenNerf: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. In *International Conference on Learning Representations*, 2024. 1, 2

[4] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision (ECCV)*, pages 540–557, 2022. 1

[5] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2. Minneapolis, Minnesota, 2019. 1

[6] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19729–19739, 2023. 1

[7] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21530–21539, 2024. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 1

[9] Hyunjee Lee, Youngsik Yun, Jeongmin Bae, Seoha Kim, and Youngjung Uh. Rethinking open-vocabulary segmentation of radiance fields in 3d space. *arXiv preprint arXiv:2408.07416*, 2024. 4

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:/1907.11692*, 2019. 1

[11] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view syn-

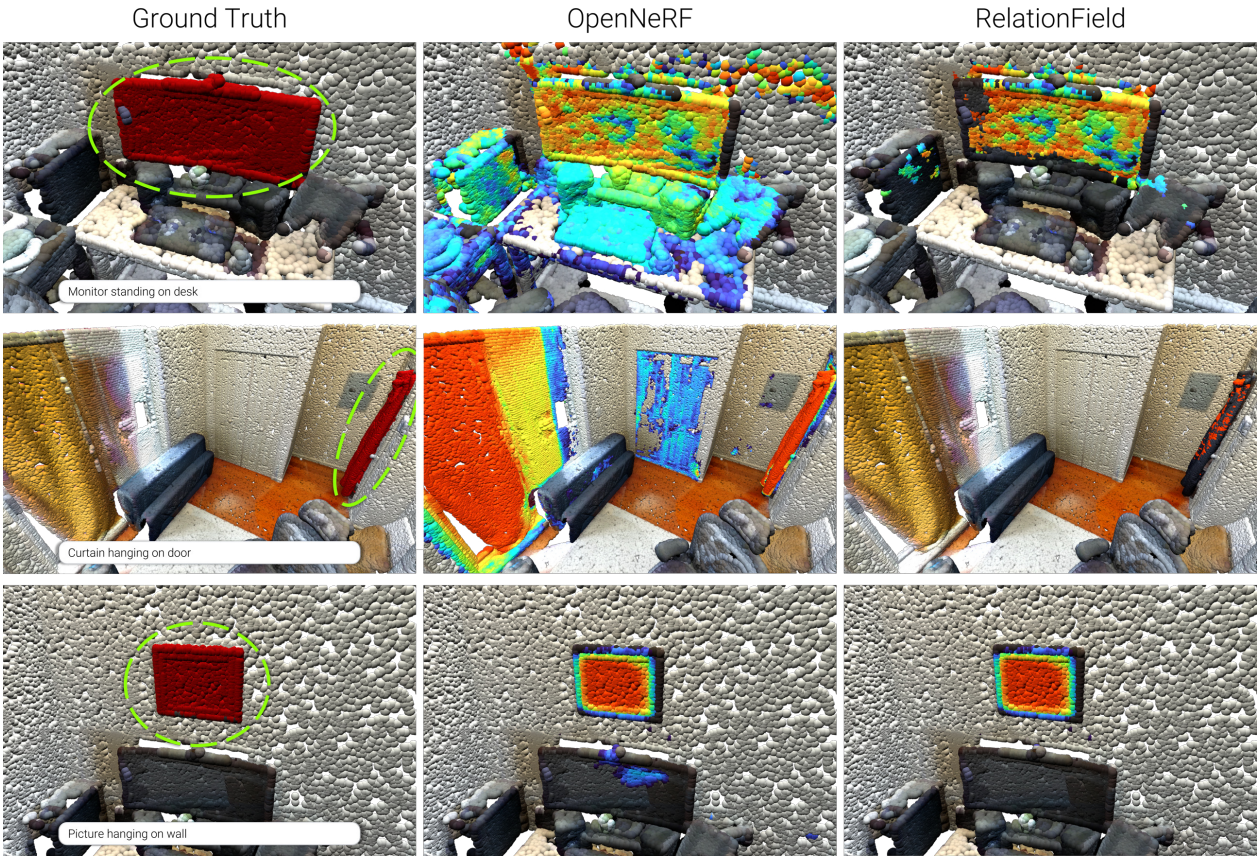

Figure 6. **Releationship-guided 3D Instance Segmentation.** We compare OpenNeRF with RelationField for relationship-guided 3D instance segmentation. While OpenNeRF produces many false positives because it gets confused with compositional queries arising from the *bag-of-words* behavior of CLIP. Meanwhile RelationField uses the object information together with the relationship information from the prompt to accurately filter predictions that only correspond to the subject in the prompt.

thesis. In *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer International Publishing, 2020. 4

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1, 3

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. CLIP. 1

[14] Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. jina-embeddings-v3: Multilingual embeddings with task lora, 2023. 1

[15] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 1

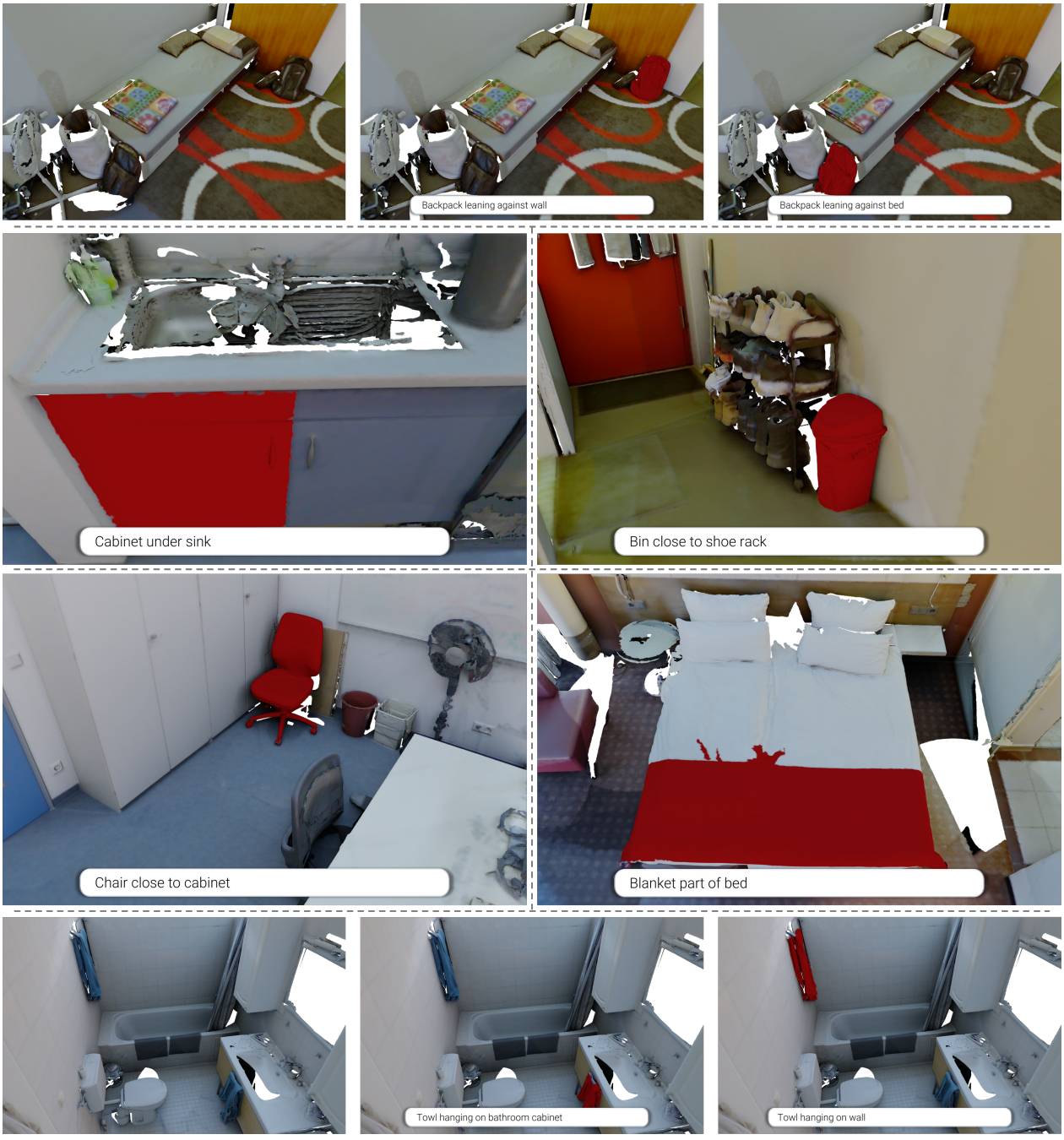

Figure 7. **Releation-guided 3D Instance Segmentation Task Overview.** We visualize a few annotated segments from our labeled benchmark on Scannet++ together with annotated relationship prompts. We focus on objects which appear multiple times in the scene, but that can be uniquely referenced by a relationship prompt.