Appendix

This document provides supplementary materials for *Weakly Supervised 3D Open-vocabulary Segmentation* in implementation details (Appendix A), more ablations (Appendix B), more evaluations (Appendix C), and more results (Appendix D).

A Implementation Details

A.1 Model Architecture

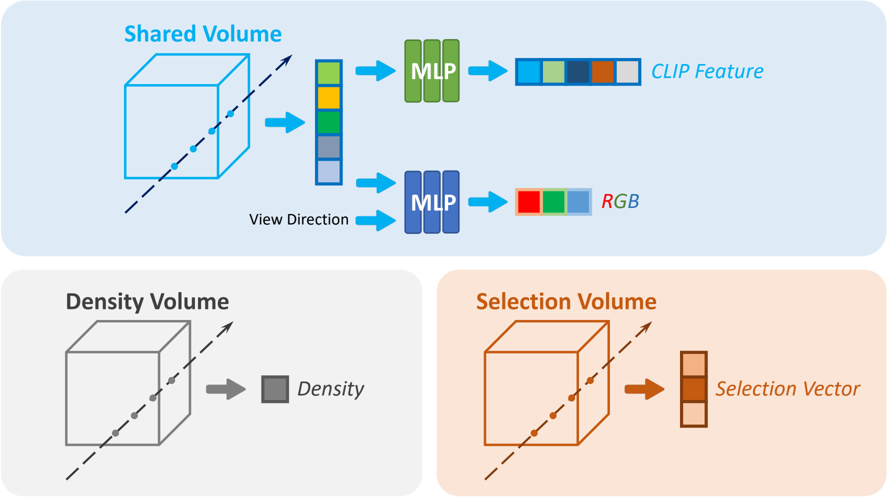

Figure 1: **Model architecture.**

We use TensoRF [1] as our base NeRF architecture for efficiency, the plane size is also the same as the default setting of TensoRF. The RGB and CLIP feature branches share the same volume and use the same intermediate features. The selection volume and density volume are two other independent volumes. We directly use the features extracted from the selection volume and density volume as the selection vector and the density value, as they have low dimensions and are view-independent. We use the original MLP architecture in TensoRF to extract the view-dependent RGB value and use another MLP which discards view direction input to extract the rendered CLIP feature. The architecture is illustrated in Fig. 1.

A.2 Hyperparameters

We set $\tau = 0.2$ to get the shaper segmentation probability distribution $\acute{P}$. The offset b is set to 0.7 to measure the similarities of the DINO features, meaning that two DINO features are considered similar if their cosine similarity is larger than 0.7, and different if less than 0.7. We use 3 scales of CLIP features, and the patch sizes of each scale are set as $s/5$, $s/7$, and $s/10$, where s is the smaller value in the width and height of the input image I. In the ablation studies, we use $s/7$ as the patch size of the single-scale CLIP feature input. The weights associated with similar and dissimilar DINO features in $\mathcal{L}_{FDA}$ are set as $\lambda_{pos} = 200$ and $\lambda_{neg} = 0.2$ by default. In certain scenes, we find that setting λ_{neg} to 0.22 or 0.18 can produce better results. We use ViT-B/16 CLIP model to extract the image and text features and use version 1 dino_vitb8 model to extract the DINO features because it employs the smallest downsampling factor of 8 which is advantageous for high-precision segmentation.

A.3 Training

To reconstruct a NeRF from multiview images of a scene, we follow the same training settings as TensoRF. For segmentation training, we train the model for 15k iterations. In the first 5k iterations,

Table 1: **Dataset.** We list the collected 10 scenes and the corresponding text labels. The background labels are in *Italic font*.

Scene	**Text Labels**
bed	red bag, black leather shoe, banana, hand, camera, *white sheet*
sofa	a stack of UNO cards, a red Nintendo Switch joy-con controller, Pikachu, Gundam, Xbox wireless controller, *grey sofa*
lawn	red apple, New York Yankees cap, stapler, black headphone, hand soap, *green lawn*
room	shrilling chicken, weaving basket, rabbit, dinosaur, baseball, *wood wall*
bench	Portuguese egg tart, orange cat, green grape, mini offroad car, dressing doll, *pebbled concrete wall*, *wood*
table	a wooden ukulele, a beige mug, a GPU card with fans, a black Nike shoe, a Hatsune Miku statue, *lime wall*
office desk	the book of The Unbearable Lightness of Being, a can of red bull drink, a white keyboard, a pack of pocket tissues, *desktop*, *blue partition*
blue sofa	a bottle of perfume, sunglasses, a squirrel pig doll, a JBL bluetooth speaker, an aircon controller, *blue-grey sofa*
snacks	Coke Cola, orange juice drink, calculator, pitaya, Glico Pocky chocolate, biscuits sticks box, *desktop*
covered desk	Winnie-the-Pooh, Dove body wash, gerbera, electric shaver, canned chili sauce, *desktop*

we freeze the shared volume and density volume, and train the selection volume and the CLIP feature branch. For the rest 10k iterations, we further finetune the shared volume and the RGB branch. We use Adam optimizer with $betas = (0.9, 0.99)$. The learning rates for training the volume and MLP branch are respectively set to 0.02 and $1e-4$. For finetuning the volume and the MLP, the learning rates are set to $5e-3$ and $5e-5$. We also employ a learning rate decay with a factor of 0.1.

The multi-scale pixel-level CLIP features of training views are pre-computed before training and the DINO features are computed with sampled patches on the fly during training. When computing $\mathcal{L}_{supervision}$ and $\mathcal{L}_{RDA}$, we randomly sample rays with a batch size of 4096. When computing $\mathcal{L}_{FDA}$ we randomly sample patches of size 256×256 with a batch size of 8. We use a downsampling factor of 8 when sampling rays and a factor of 5 when sampling patches. The model is trained on an NVIDIA A5000 GPU with 24G memory for $\sim$1h30min for each scene.

A.4 Dataset

We capture 10 scenes using smartphones and use Colmap [2] to extract camera parameters for each image. We capture $20 \sim 30$ images for each scene and the resolution of each image is 4032×3024. We follow the data structure of LLFF [3]. We manually annotate the segmentation maps of 5 views for each scene as the ground truth for evaluation.

We list the text labels used in our experiments in Tab. 1. Note that we sometimes choose to add a general word to describe the whole background, such as *wall*, *desktop*, following LERF [4]. The text labels in our dataset contain many long-tail classes, which can be used to fully assess the open-vocabulary capability.

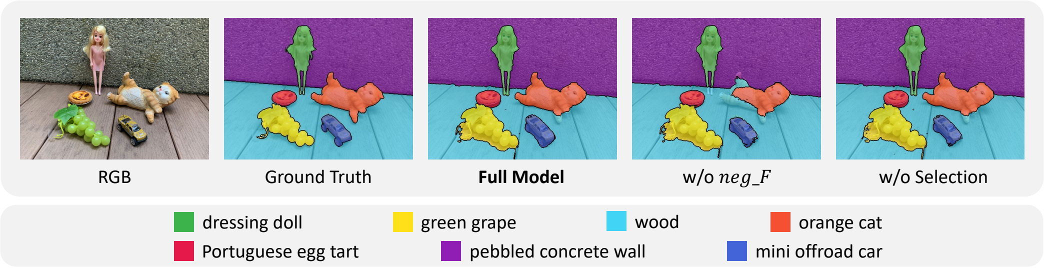

Figure 2: **More ablations.**

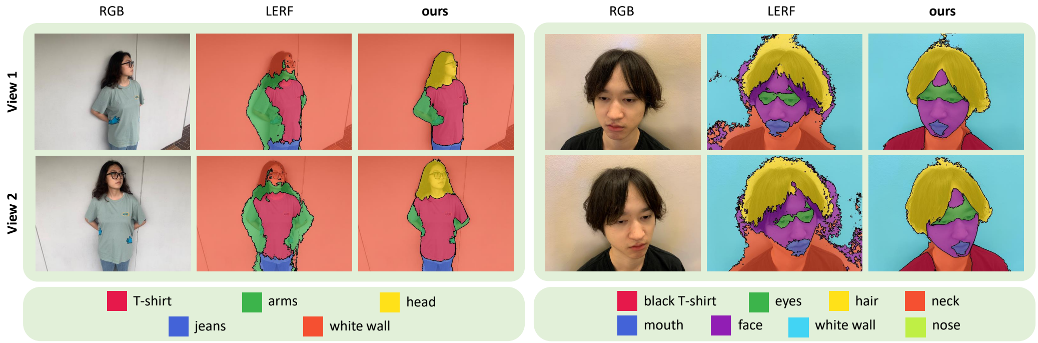

Figure 3: Evaluation on human body dataset (left). Evaluation on human head dataset (right).

B More Ablations

We perform two more ablation studies on the Selection Volume and the FDA loss, as shown in Tab. 2 and Fig. 2. Without the Selection Volume, we simply average the multi-scale CLIP features rather than learning the appropriate scale. We can see that both the mIoU score and the mAP score are inferior to the full model. We could discard the dissimilar part neg_F since dissimilar DINO features often impair the stability of the correlation loss. However, neg_F encourages different segmentation probabilities for different semantic regions and it plays a crucial role for precise object boundary extraction.

Table 2: **More Ablations.**

	mIoU	**mAP**
w/o neg_F	76.9	92.4
w/o Selection	84.8	95.3
full model	**86.2**	**95.8**

C More Evaluations

We additionally perform evaluations on human body, human head, indoor datasets with low-quality images [5, 6], and a complex scene from LERF datasets [4]. We compare with the concurrent work LERF qualitatively due to the lack of labels or the defective annotations as pointed out in [7]. We also perform experiments with different text prompts. We use the same scale level number and patch sizes in all comparisons.

Human body and head. As shown in Fig. 3, our method segments more precise parts than LERF. Specifically, LERF fails to segment the "head" in the human body and the "black T-shirt" in the human head. In contrast, our method can recognize and segment these parts correctly because our designed RDA loss addresses the ambiguity of the CLIP features effectively.

Indoor scenes with low-quality images. Fig. 4 shows experiments on the indoor datasets [5, 6], where many images are unrealistically rendered with less photorealistic appearances (as indicated in [7]) and have limited spatial resolution (640 × 480 or 1024 × 768). Due to these data constraints,

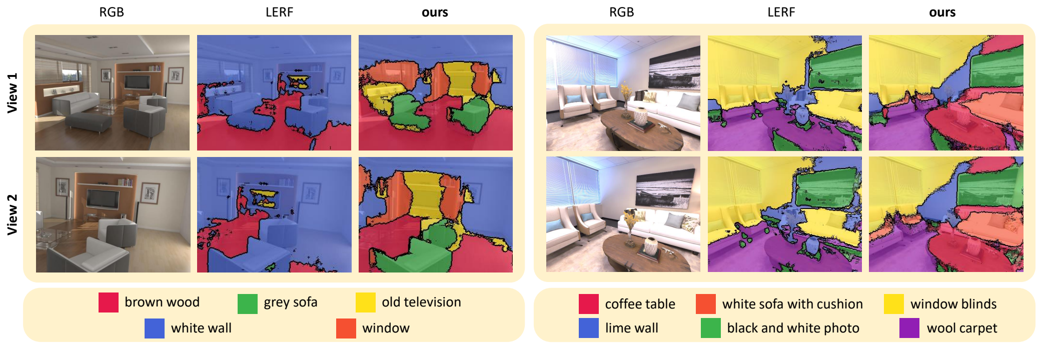

Figure 4: Evaluation on indoor datasets with lower quality images.

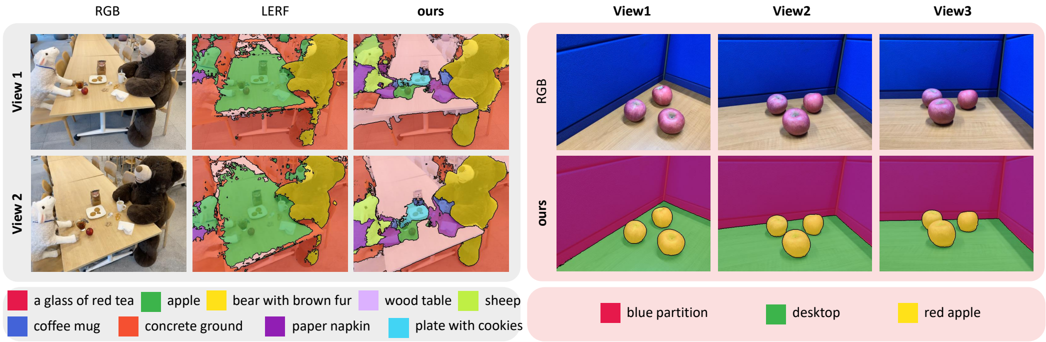

Figure 5: Evaluation on a complex scene from LERF datasets (left). Evaluation on a scene with multiple instances of a same class (right).

our method sometimes confuses labels with similar appearances. However, we can see that our method still outperforms LERF by successfully segmenting more labels.

Complex scenes. Fig. 5 (left) shows the segmentation of one challenging sample from the LERF dataset, where the scene has complex geometry as well as many objects of varying sizes. It can be observed that LERF cannot segment most objects due to the ambiguities of CLIP features while our method can segment more objects correctly with more precise boundaries. Fig. 5 (right) shows a scene with multiple instances of a same class. Since instances of the same class often share similar appearance, texture, etc., they also have similar DINO features. As a result, FDA will not mistakenly segment them into different classes. The RDA loss will further help by assigning all these instances to the same text label. In the experiment, we observed that our method successfully segments all three apples into the same class with accurate boundaries.

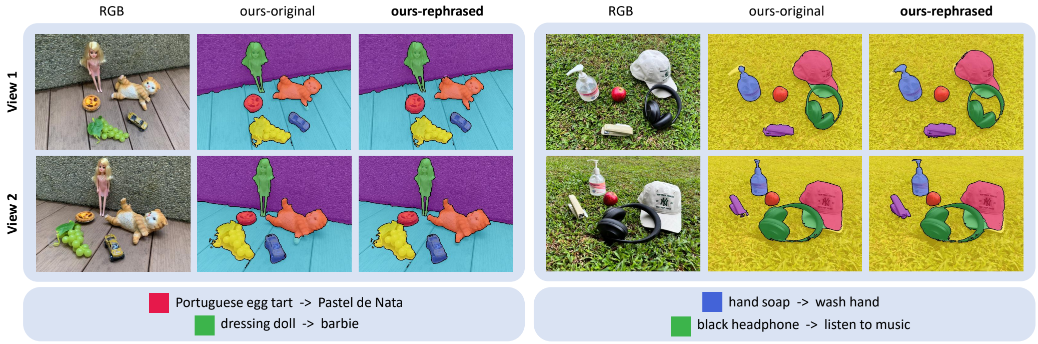
RGB
ours-original
ours-rephrased
View 1
View 2
🟥 Portuguese egg tart -> Pastel de Nata
🟩 dressing doll -> barbie
RGB
ours-original
ours-rephrased
🟦 hand soap -> wash hand
🟩 black headphone -> listen to music

Figure 6: Evaluation on rephrased texts.

Segmentation with different text prompts. We conduct experiments to segment scenes with different text prompts. In the experiments, we replaced the original texts with different languages (e.g., Portuguese egg tart -> Pastel de Nata), names (e.g., dressing doll -> Barbie), and actions (e.g., hand soap -> wash hand, black headphone -> listen to music). As Fig. 6 shows, with the rephrased text prompts, our method can still segment the scenes reliably. The experiments are well aligned with the quantitative experiments as shown in Tab. 3.

Table 3: Evaluation on rephrased texts.

	mIoU	mAP	mIoU	mAP
original	88.2	97.3	89.3	96.3
rephrased	89.3	97.2	88.4	96.6

D More Results

We show more segmentation visualizations of our method in Fig. 7 (bed), Fig. 8 (sofa), Fig. 9 (lawn), Fig. 10 (room), Fig. 11 (bench), Fig. 12 (table), Fig. 13 (office desk), Fig. 14 (blue sofa), Fig. 15 (snacks), and Fig. 16 (covered desk). The quantitative results are listed in Tab. 4.

Table 4: **Quantitative results.**

bed		*sofa*		*lawn*		*room*		*bench*	
mIoU	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**
89.5	96.7	74.0	91.6	88.2	97.3	92.8	98.9	89.3	96.3
table		*office desk*		*blue sofa*		*snacks*		*covered desk*	
mIoU	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**	**mIoU**	**mAP**
88.8	96.5	91.7	96.2	82.8	97.7	95.8	99.1	88.6	97.2

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 2022. 1

[2] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[3] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 2

[4] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *arXiv preprint arXiv:2303.09553*, 2023. 2, 3

[5] 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. 3

[6] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 3

[7] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 3

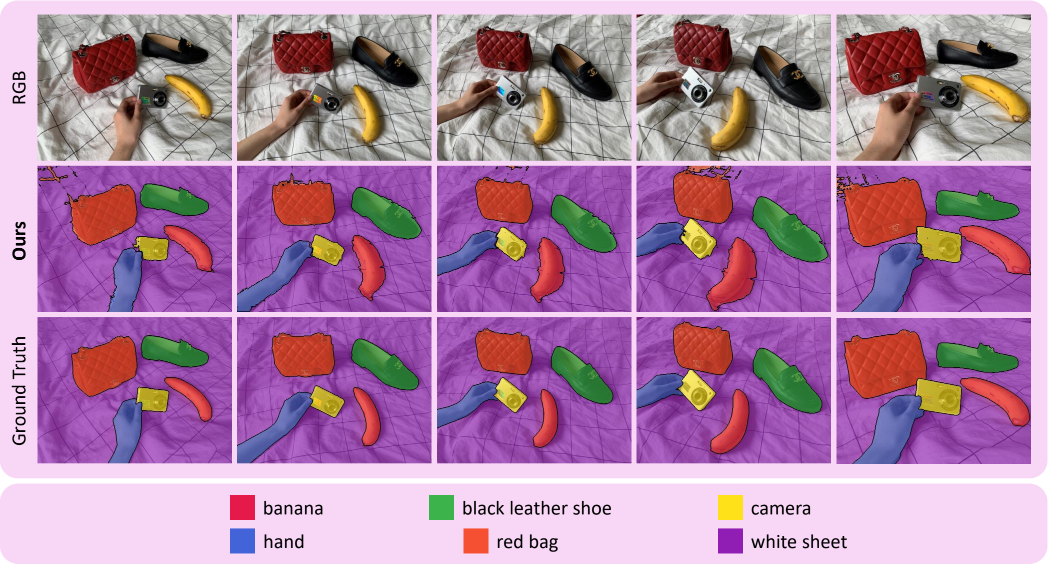

Figure 7: **bed.**

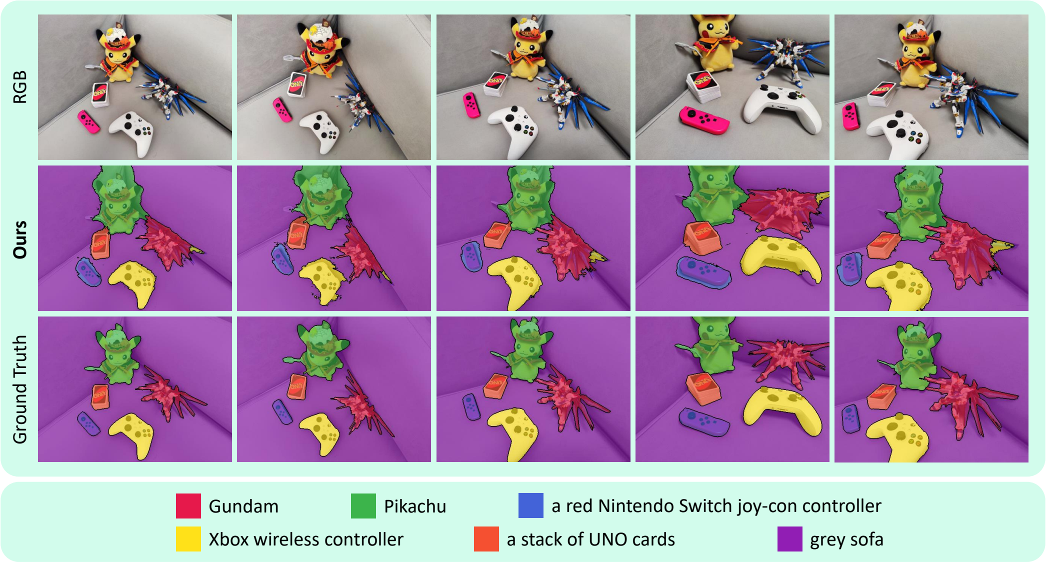

Figure 8: **sofa.**

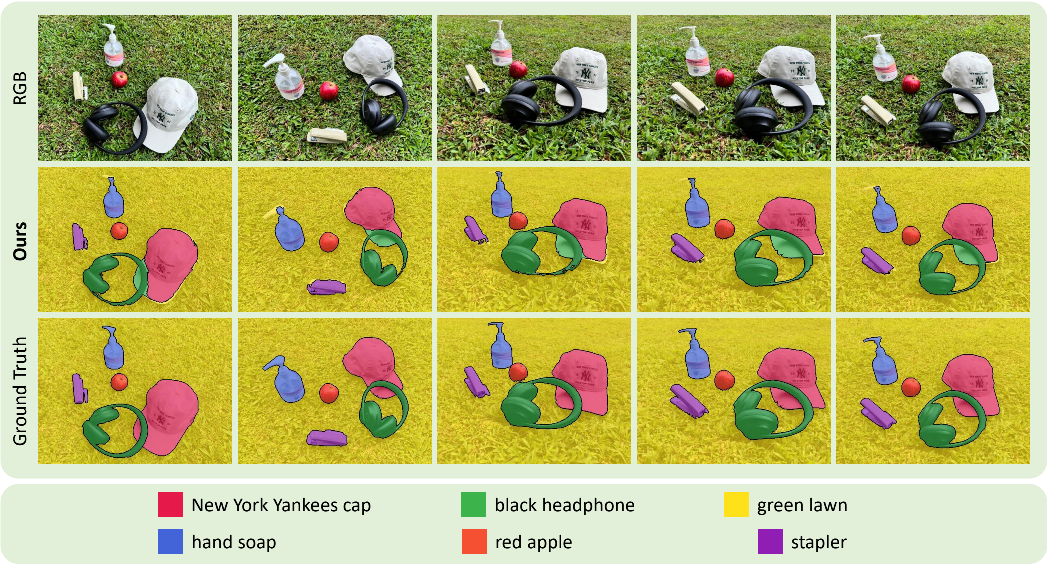

Figure 9: **lawn.**

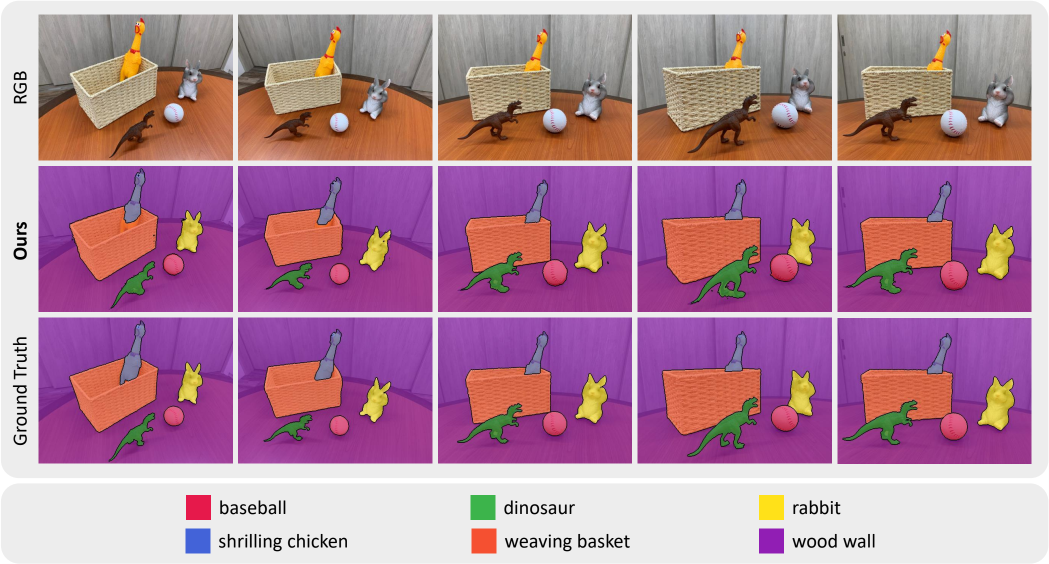

Figure 10: **room.**

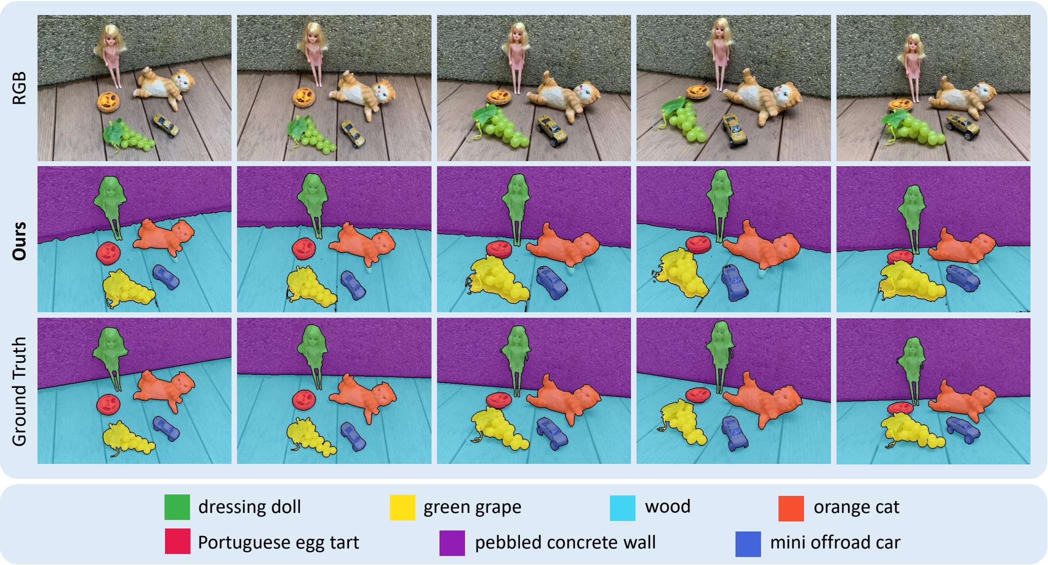

Figure 11: **bench.**

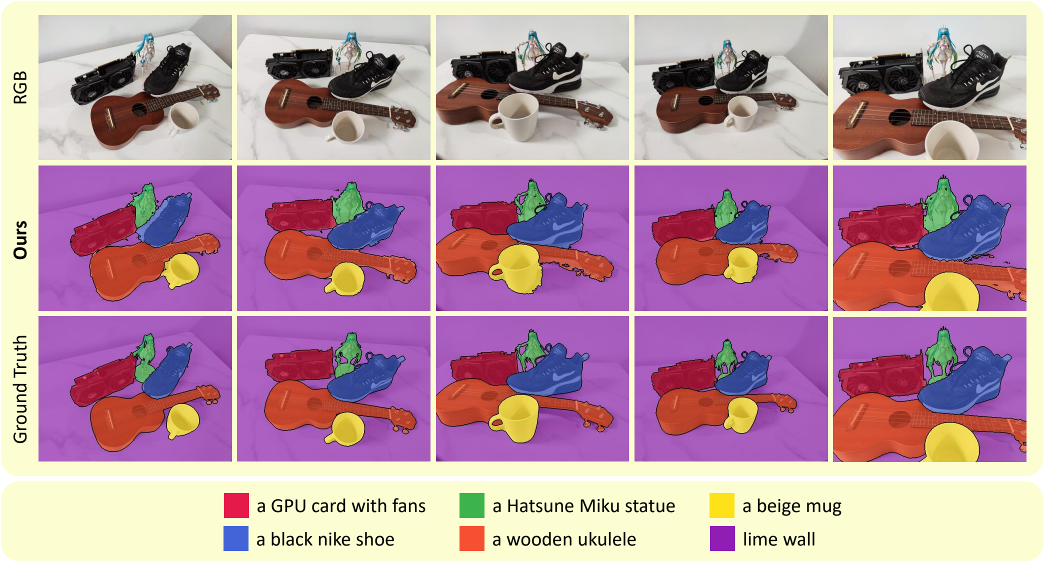

Figure 12: **table.**

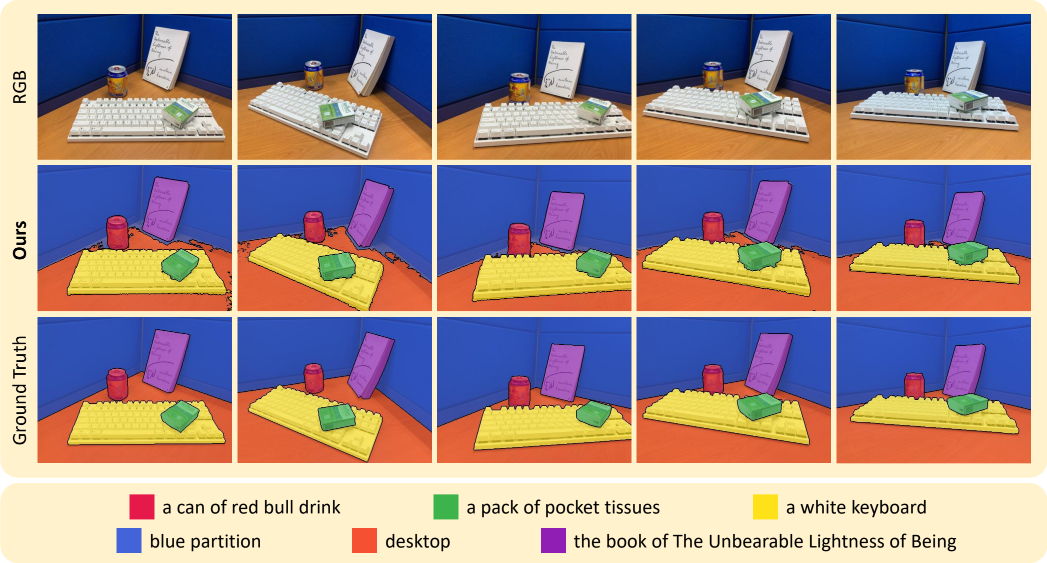
RGB
Ours
Ground Truth
🟥 a can of red bull drink
🟩 a pack of pocket tissues
🟨 a white keyboard
🟦 blue partition
🟧 desktop
🟪 the book of The Unbearable Lightness of Being

Figure 13: **office desk.**

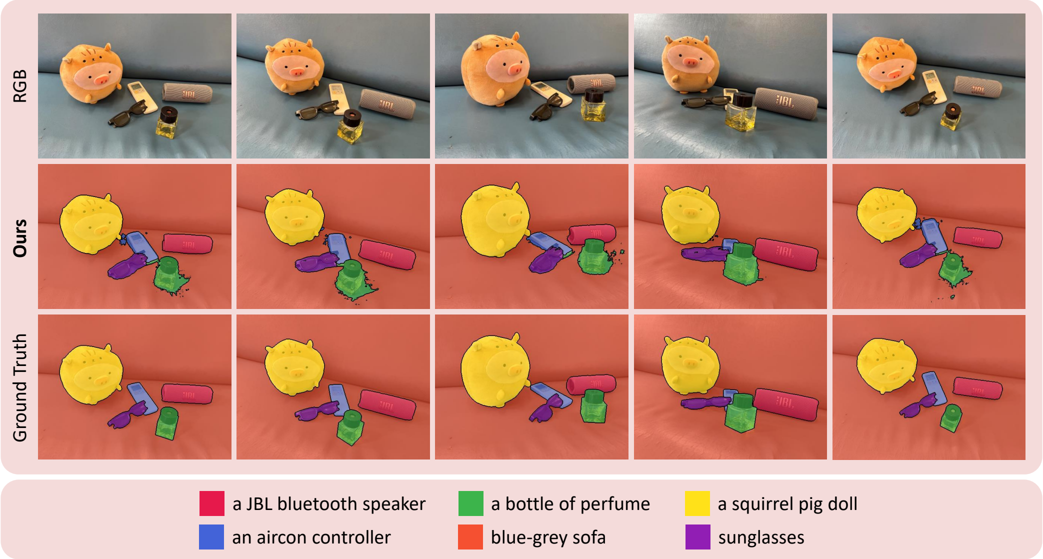

Figure 14: **blue sofa.**

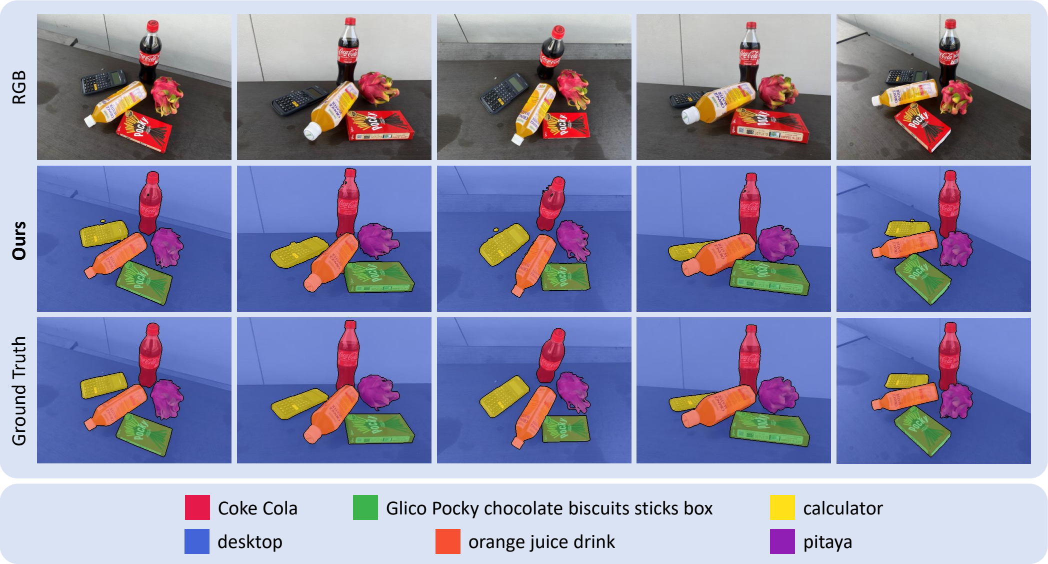

Figure 15: **snacks.**

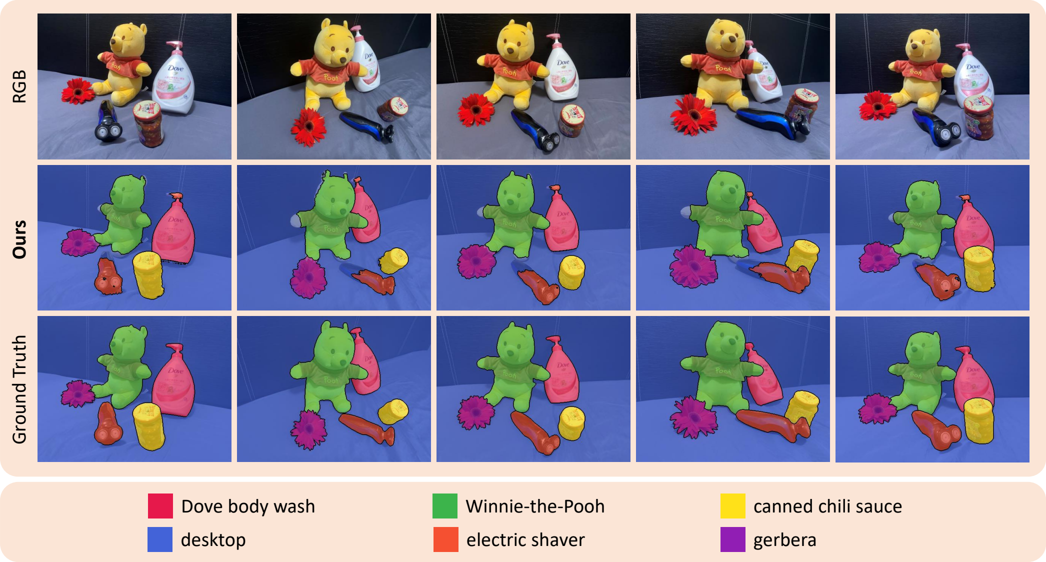

Figure 16: **covered desk.**