Prompt Then Ground: Task-Aware Scene Understanding via Online Neural Implicit Mapping | Supplementary Material

1 Detailed Pipeline

Figure S1 illustrates the ProGround pipeline, which consists of two main stages: prompt-driven feature extraction and online grounding with a neural implicit field.

In the **feature extraction stage**, we begin by defining a set of task-relevant labels through text prompts (e.g., *couch*, *bed*, *picture*). These prompts are passed to the YOLO-World [1] detector alongside the RGB input, producing bounding boxes, class labels, and associated confidence scores. To ensure task relevance and reliability, detections below a confidence threshold (0.4 in our experiments) are discarded. The remaining instances are further refined using FastSAM [3] to obtain pixel-wise segmentation masks, which are then associated with bounding boxes to extract semantic features and confidences.

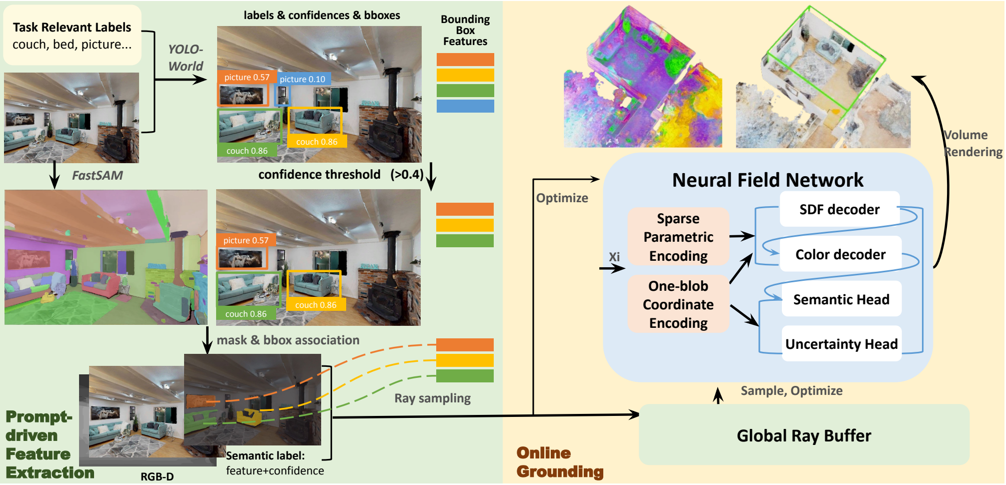

Figure S1: **Prompt-then-Ground Pipeline.** The feature extraction process generates pixel-wise semantic features along with scores from visual observations. Afterwards, the neural field is optimized with sparse samples from instant observations and replayed experiences.

In the **online grounding stage**, semantic features, along with RGB-D and pose information, are used to sample rays from the current observation. These rays are stored in a global replay buffer and used to supervise a neural implicit field network. The network consists of one-blob coordinate encoding and a sparse parametric encoding module [2], followed by decoders for signed distance function (SDF), color, semantics, and uncertainty. During optimization, the model performs volume rendering over sampled rays to reconstruct scene properties. The uncertainty head is extended for predicting feature scales, with the last channel of its logits distinguishing supervised from unsupervised regions, while

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

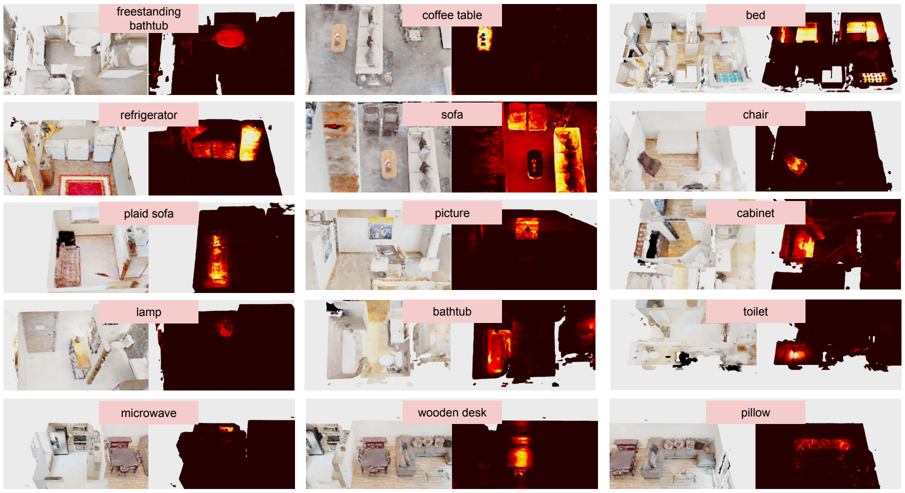

Figure S2: **Examples of successful query localization.** Each visualized result highlights the model's ability to accurately localize both common and fine-grained object categories.

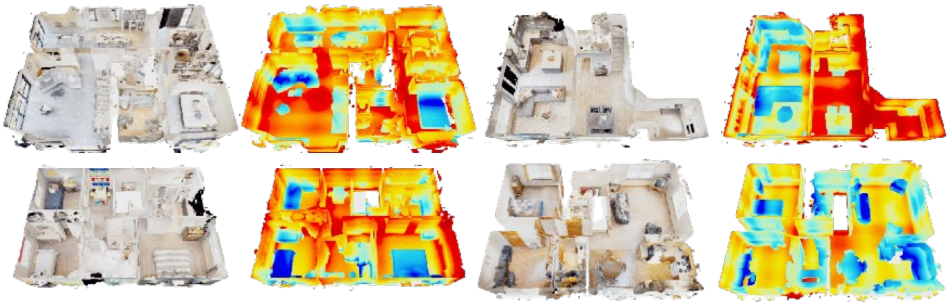

Figure S3: **Semantic uncertainty estimation across scenes.** For each pair, left: reconstructed RGB scene, right: predicted semantic uncertainty. High uncertainty (red) often aligns with unsupervised regions. These maps help identify where supervision is lacking and improve downstream query reliability.

the semantic head operates on fused positional and color features to promote robust convergence under sparse supervision.

2 Extra Visualization Examples

2.1 Query Visualization

To qualitatively evaluate ProGround's ability to support open-vocabulary queries, we visualize examples of text-based object queries grounded within the reconstructed 3D scenes. These visualizations demonstrate how well the model aligns semantic features with textual descriptions under varying conditions and categories.

2.2 Uncertainty Visualization

In this section, we visualize the predicted semantic uncertainty across multiple scenes. These uncertainty maps are inferred by the last channel of our model feature scale head and help identify regions lacking semantic supervision or exhibiting ambiguous predictions. Such visualizations provide insights into where the representation may be unreliable.

Table S1: Computational cost of each module (ms/step)

Feature extraction	Segmentation	Forward (color&geo)	Forward (semantics)	Backward
78.75	93.73	120.82	162.59	304.29

Table S2: Top-K accuracy (↑) with different weighting and filtering strategies.

Weighting	Filtering	top$_1$	top$_5$	top$_{10}$	top$_{20}$	top$_{50}$
✗	✗	13.31	29.29	37.17	49.40	68.31
✓	✗	28.55	45.61	54.26	63.65	**76.88**
✓	✓	**32.24**	**48.74**	**56.89**	**65.24**	76.53

Table S3: Effect of irrelevant prompts on the Top-K Accuracy (↑)

Prompt Ratio	top$_1$	top$_5$	top$_{10}$	top$_{20}$	top$_{50}$
Task Only	**32.24**	**48.74**	**56.89**	**65.24**	**76.53**
+ 20%	28.11	43.40	52.14	61.93	75.27
+ 50%	27.22	40.70	49.32	58.95	73.16
+ 100%	26.45	39.17	47.30	56.75	70.87

Table S4: Top-K accuracies given different choices of the embedding space for semantic decoding.

Settings	w/o Coord					w/ Coord				
	top$_1$	top$_5$	top$_{10}$	top$_{20}$	top$_{50}$	top$_1$	top$_5$	top$_{10}$	top$_{20}$	top$_{50}$
Color	19.89	35.51	44.31	53.38	66.09	**32.24**	**48.74**	**56.89**	**65.24**	**76.53**
Geo	12.82	25.97	34.19	43.64	59.40	29.50	47.76	56.15	64.32	74.57
Param	10.45	21.55	29.16	37.93	52.65	30.06	46.43	55.19	63.86	75.90
Coord	–	–	–	–	–	28.11	45.01	53.81	62.58	73.01

3 Implementation Details

We set the semantic confidence threshold to 0.4 across all experiments. We store 1 keyframe per 5 frames and store 5% of all valid rays per keyframe for supervision.

For training the neural field, we apply a weighted multi-loss function with the following coefficients: 5.0 for RGB reconstruction loss, 0.8 for semantic feature loss, 0.1 for depth loss, 1000.0 for SDF regression loss, 10.0 for feature scale classification, and 1e-5 for a smoothness regularization loss, applied to 32 sampled points within a voxel radius of 0.1 and a truncation margin of 0.05.

Our decoder architecture uses a geometry feature dimension of 15, with 2-layer MLPs for the SDF, color, and semantic branches. Each branch has a hidden dimension of 32, and the semantic decoder receives shared features from both color and positional encodings.

4 Limitations

One major limitation is the efficiency of NeRF-SLAM. As shown in Tab. S1, the forward and backward processes take around 0.6 seconds for each step. The modules can be further accelerated with a hybrid map representation to avoid the computation of density in free space. Meanwhile, the segmentation also restricts the system from efficient deployment. As the system only requires sharp segmentation given prompted bounding boxes, lightweight segmentation models may further reduce the computational cost.

In addition, as shown in Tab. S2, S3, and S4, the ablation results given a different Yolo-World architecture (Yolo-World-L) share the similar trend compared to the results in the main paper (Yolo-World-XL), where prompt-then-ground and proper feature sharing can lead to better results. However, the differences in accuracy between the two architectures are large. The final results are highly correlated with the selected perception model.

References

[1] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:2401.17270*, 2024.

[2] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

[3] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.