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

Mengying Lin
College of Computing
Georgia Institute of Technology
Atlanta, GA 30308
mlin365@gatech.edu

Zijia Kuang
Institute for Al Industry Research (AIR), Tsinghua University
Beijing, China
kzj18@tsinghua.org.cn

Ting Li
Institute for Al Industry Research (AIR), Tsinghua University
Beijing, China
ting2025.li@tum.de

Zike Yan
Institute for Al Industry Research (AIR), Tsinghua University
Beijing, China
zike.yan@pku.edu.cn

Guyue Zhou
Institute for Al Industry Research (AIR), Tsinghua University
Beijing, China
zhouguyue@air.tsinghua.edu.cn

Abstract

Open-vocabulary scene understanding is crucial for robotic applications, involving locating targets from 3D semantic scene representations given queries. However, existing mapping approaches often focus on task-agnostic representations, suffering from inaccurate semantic supervision due to noisy and ambiguous perception. We introduce ProGround, a prompt-then-ground framework for online neural implicit mapping that reshapes the data distribution to prioritize task-relevant and high-confidence semantic features. We exploit in-network aggregation of local-global feature pyramids with sufficient context information. To ensure fast optimization with accurate reasoning, we probe semantics from concatenated features of positional and color embedding and employ a selective experience replay mechanism for continual learning with forgetting avoidance. Evaluated on Habitat-Sem, ProGround achieves a +4.36% improvement in Top-1 semantic accuracy over state-of-the-art methods while maintaining memory efficiency. Applications for robotic navigation reveal great potentials with the proposed paradigm.

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

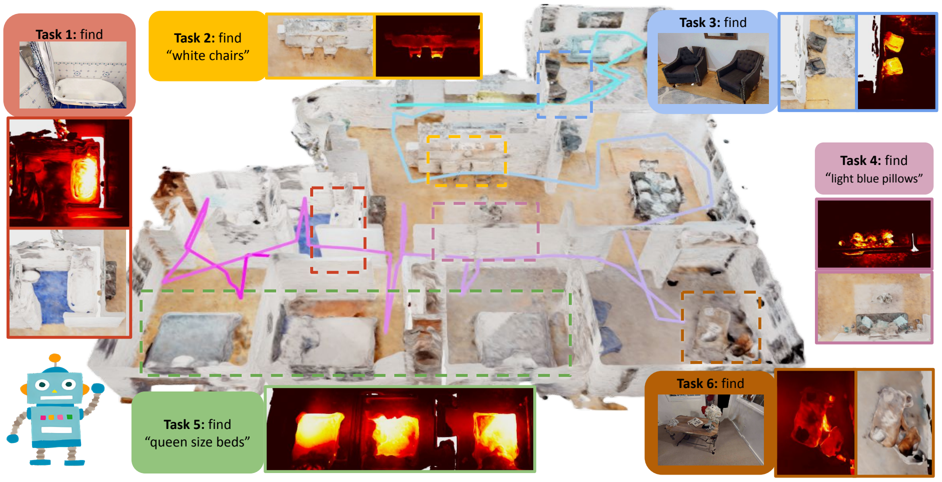


Figure 1: **ProGround constructs a queryable task-oriented neural map.** It grounds features with the awareness of task-relevance and confidence and supports query with both images and texts, suitable for task-driven applications like robotic navigation and manipulation.

1 Introduction

In recent years, scene understanding has increasingly focused on 3D visual grounding, which aims to locate specific objects in environments that semantically correspond to given queries. This involves constructing queryable map representations that capture environmental semantics, allowing robots to reason about their surroundings in an interpretable manner. Traditional grid-map-based methods [2, 20, 36] typically rely on closed-vocabulary representations, where objects and concepts are restricted to a fixed set of categories. This limitation hinders generalization to out-of-domain targets, synonyms, and diverse linguistic expressions. To address this, recent advancements have shifted toward open-vocabulary representations, leveraging progress in vision-language models [1, 23, 12]. Unlike their closed-vocabulary counterparts, open-vocabulary representations leverage visual features from a rich, language-aligned semantic space, allowing for flexible storage of various categories and multi-modal querying. This enables robots to interpret and handle a broader range of human instructions effectively. Ideally, we aim to construct scene representations that capture all semantic details with high fidelity. **A major challenge in achieving this is perception ambiguity due to occlusions and limited viewpoints, which may lead to incorrect semantics.** In offline scene construction, where all frames are available simultaneously, iterative and computationally intensive refinements can be applied to ensure a cleaner representation. However, this becomes infeasible given streaming data as computational resources and processing time are limited: systems must process streaming data within strict time constraints while handling incomplete or obstructed views, making 3D visual grounding more difficult. As illustrated in Fig. 1, Considering that grounding is often task-driven, particularly in applications like robotic navigation and manipulation, prioritizing task-relevant information while reducing focus on irrelevant and ambiguous data leads to a goal-aligned representation that is effective for downstream applications with promising efficiency.

Therefore, we propose a *prompt-then-ground* paradigm for constructing online, task-oriented map representations. To achieve this, we specify task-relevant features guided by prompts, which reshape the raw observations into a task-oriented one for grounding. Storing such high-dimensional features on-the-fly raises challenges regarding computational and memory efficiency. Compared to the discrete representations [8, 31], neural fields [25, 35, 11, 6] reveal nice properties by encoding color, geometry, and semantics implicitly and maintain a fixed size regardless of the changing scene scales. To ensure efficiency and accuracy during the online optimization, we probe semantics upon a feature space with preliminary cues from positional and color information. A replay-based continual learning strategy is deployed for online optimization that stores a sparse set of samples to ensure adaptation to new observations while preserving past knowledge. The prompted semantic features with task relevance scores supervise the neural field as a weighted grounding of task-relevant information. Our contributions can be summarized as follows:

- We propose a prompt-then-ground paradigm for task-aware scene understanding. The open-vocabulary semantic features with task-relevancy form a compact and queryable representation.
- We exploit a selective experience replay solution to balance between reconstruction accuracy and efficiency given the task-oriented features.
- We obtain a continuous representation with promising memory-efficiency and accuracy. Appearance, geometry, and task-relevant semantics are encoded compactly, supporting semantic navigation given different modalities.

2 Related Work

2.1 Foundation Models for Semantic Mapping

Open-vocabulary representation has emerged as a prominent trend in semantic mapping due to its ability to generalize beyond closed-set category definitions, where vision-language models (VLMs) play a crucial role in providing modality-aligned features. To deal with image-wise features [23, 16, 15], a widely adopted mapping approach is segment-then-encode [31, 19, 28], where images are first segmented using class-agnostic models [12, 37] before being processed by the VLMs. HOV-SG [31] constructs an open-vocabulary 3D scene graph by associating segment-based features across frames. CLIO [19] incrementally builds a 3D representation by clustering 2D segment-based features into nodes, filtering task-irrelevant information with information bottleneck [29]. O2V-Map [28] grounds segment-based features into an open-vocabulary field to maintain spatial and semantic consistency. However, this approach is susceptible to issues introduced by segmentation models, over-segmentation for instance, which may degrade performance.

An alternative is direct mapping with finer-grained VLM features from open vocabulary models, bypassing reliance on segmentation models. These features can be categorized as pixel-level [14, 5] or region-level [40, 7, 4, 41]. The features from these models eliminate the limitations introduced by segmentation models and provide more reliable semantics for mapping. Recent work of Open-NeRF [6] leverages the finer-grained VLM features with a NeRF-based representation, but relies on iterative optimization over the entire dataset in an offline setting. We take a step further to ground task-relevant features on-the-fly to remove the redundancy while maintaining an online system.

2.2 Trade-offs in Feature Grounding

Early efforts in grounding modality-aligned features primarily focus on offline mapping [3, 9, 11, 6], where observations are available all at once and representations are constructed prior to deployment. These methods benefit from extensive iterative refinement, reprocessing of past data, and global optimization of learned representations without time constraints. This leads to highly refined scene understanding but typically requires significant computational resources and processing time, limiting their practicality in resource-constrained scenarios and live deployment. In contrast, online mapping has gained attention for its ability to incrementally fuse vision-language features in a real-time budget [22, 8, 19, 33]. This imposes strict constraints on latency and memory consumption, requiring the system to balance accuracy with efficiency, where association is usually conducted in a coarse manner with efficient data fusion or marginalization.

Besides the balance between accuracy and efficiency, another key factor is to minimize the storage of cumbersome information across views. This is usually dependent on the corresponding representations, either in an explicit or implicit fashion. For explicit representations, the high dimensional features are explicitly stored in discretized primitives such as point cloud [10] or grids [9], which is typically less memory-efficient. To avoid the redundancy, sparse graph [8, 19] and dictionary [33] structures are exploited to maintain a sparse set of features that are locally consistent. Nevertheless, the local consistency requires non-trivial and manual association to establish spatial-temporal correlations. Implicit representation on the other hand, encodes features within neural implicit fields [3, 24] and optimize the parameters through differentiable rendering, enabling continuous and memory-efficient feature grounding. Given the computational constraints of online mapping and the need for efficient feature storage, we take the implicit representation and adopt a selective experience replay strategy that incrementally integrates vision-language representations into the neural field.

3 Preliminary

We take in a stream of camera poses and RGB-D images given known camera intrinsics and aim to construct a neural implicit field $\boldsymbol{\theta}$ incrementally, which maps spatial coordinates $\mathbf{x}$ to scene properties $\mathbf{y}$, including color $\mathbf{c}$, geometry s, and task-relevant semantics $\mathbf{f}$ defined by a specific grounding task $\mathcal{T}$. The optimization involves supervising these outputs using the corresponding information within the observations. The overall optimization objective of the neural field can be written as:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta})], \tag{1}$$

where $\mathcal{L}$ is the self-supervised loss through differentiable rendering specified in Sec. 4.

In offline learning, where all data $\mathcal{D}$ is available at once, we can carry out global joint optimization by minimizing the empirical risks. However, in an online setting, training the feature field using streaming data undergoes constant distribution shifts that may lead to catastrophic forgetting. To address this, past experience is usually stored in a dynamic buffer $\mathcal{M}^t$ with key frames [27] to approximate the distribution of the entire sequence. The objective is then to minimize the losses of past experience $\mathcal{M}^t$ and the instant observation $\mathcal{S}^t$ balanced by the hyperparameter β as:

$$\boldsymbol{\theta}^t = \arg\min \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}^t} \mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta}^t) + \beta \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{M}^t} \mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta}^t). \tag{2}$$

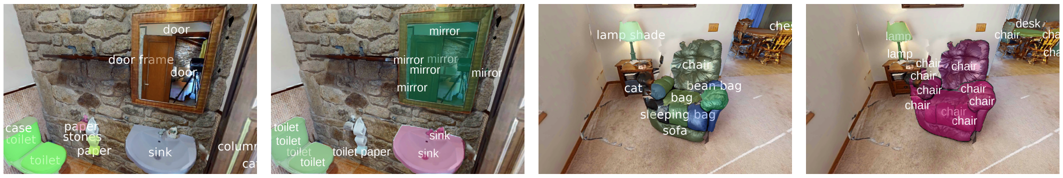
For feature grounding with vanilla neural fields, methods [38, 11] usually treat all observations equally regardless of their relevance or reliability. This can lead to the accumulation of ambiguous semantics and degrade the learned representation, as spurious or uncertain features may be reinforced over time. To mitigate the influence of low-relevance and uncertain information, we propose a *prompt-then-ground* framework, which reshapes the data distribution to be highly task-focused, which is equivalent to prioritizing data points with high relevance and confidence. The relevance can usually be acquired from the foundation models controlled by provided prompts, indicating the confidence of a detected object w.r.t. the prompted labels. Therefore, the provided confidence scores $c_{\mathcal{T}}(\mathbf{x})$ controlled by tasks form a task-oriented distribution given weighted replay buffer $\mathcal{M}_{\mathcal{T}}$ as:

$$P(\mathbf{x},\mathbf{f}|\mathcal{D}_{\mathcal{M}_{\mathcal{T}}}) = c_{\mathcal{T}}(\mathbf{x})\mathbb{I}(c_{\mathcal{T}}(\mathbf{x}) > \lambda_{\mathcal{T}})P(\mathbf{x},\mathbf{f}|\mathcal{D}_{\mathcal{M}}), \tag{3}$$

where $c_{\mathcal{T}}(\mathbf{x})$ represents the semantic confidence of a sample $(\mathbf{x},\mathbf{f})$, $\mathbb{I}$ is an indicator function, and $\mathbb{I}(c_{\mathcal{T}}(\mathbf{x}) > \lambda_{\mathcal{T}})$ filters out low-confidence samples below the threshold. The score $c_{\mathcal{T}}$ ensures that task-relevant samples are emphasized.

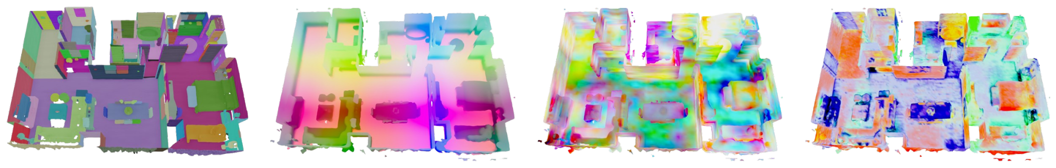
4 Methods

Our pipeline consists of two stages: prompt-driven feature extraction (Sec.4.1) and online grounding via implicit neural fields, where the architecture is elaborated in Sec. 4.2, followed by the selective replay-based optimization process in Sec. 4.3.



(a) Segment-then-encode (b) Ours (c) Segment-then-encode (d) Ours

Figure 2: **Issues of segment-then-encode paradigm.** Segment-then-encode paradigm often fails to correct mislabeling through manually fused local and global features and struggles to provide reliable semantics when instances are over-segmented.



(a) Instance segmentation (b) Coordinate embedding (c) Parametric embedding (d) Color embedding

Figure 3: **PCA results of features from different modules.** Coordinate embedding encodes relative height and spatial positions. Optimizable features like parametric and color embeddings provide finer details, with color features closer aligning with ground truth segmentations.

4.1 Hierarchical Feature Extraction with Task Relevance

3D visual grounding requires features with fine-grained details and broad context information. As discussed in Sec. 2.1, the *segment-then-encode* [10, 31, 19] paradigm attempts to fuse local instance features with global image context with a heuristically selected weight through careful manual tuning. Even with perfect segmentation, local features are often biased and can hardly be altered as the weight is controlled by the local-global similarities. For instance, a mirror shown in Fig. 2a is misclassified as a door and door frame due to its reflection. These errors propagate during multi-frame fusion, compromising the reliability of the final map. Compounding the issue, class-agnostic segmentation often suffers from over-segmentation caused by complex textures and lighting conditions. This fragments instance-level features, making local context misleading. As shown in Fig. 2c, a black chair is erroneously divided into multiple segments with noisy labels. Such excessive segmentation highly degrades semantic consistency across views.

In contrast, inherently fusing local and global context within the network structure offers a more cohesive alternative. Rather than relying solely on isolated local features, we seek features that capture and aggregate multi-scale contexts from visual inputs. In-network aggregation broadens the receptive field of instance semantics by incorporating global image context through trainable convolutional architectures. Specifically, we exploit YOLO-World [4] that integrates a Path Aggregation Network (PAN)[18] to enhance feature propagation across different scales. As demonstrated in Fig. 2b and 2d, the model infers instance-wise semantics and aggregate local-global information directly from the image without manual efforts. With the pre-defined task $\mathcal{T}$ as prompts, the model outputs the bounding boxes and the corresponding multi-scale features $\mathbf{f}$ with task-relevance scores $c_{\mathcal{T}}$. Prompted segments can then be obtained with FastSAM [37] for dense semantic features with clear boundaries. We will show in the experiment section that the prompted fashion effectively mitigate the low-confident and irrelevant features for task-focused supervision.

4.2 Shared Embeddings to Balance Acuracy-Efficiency for Implicit Neural Mapping

We maintain a unified and compact neural representation of the scene geometry, appearance, and semantics with promising memory efficiency. A key challenge lies in fast convergence given the online computational budget. Our representation follows a coordinate-based encoder-decoder structure, where the primary consideration is selecting a proper hidden space for accurate and efficient semantic decoding. In practice, we follow CO-SLAM [30] that utilizes one-blob coordinate encoding $\gamma(\mathbf{x})$ and hash-based parametric encoding $\varsigma(\mathbf{x})$ for faster convergence. A two-layer MLP is taken as the geometry decoder that takes in the concatenated features $[\gamma(\mathbf{x}), \varsigma(\mathbf{x})]$ and predicts SDF s and a geometric feature $\mathbf{f_g}$. Another two-layer MLP is taken as the color decoder that takes in the concatenated features $[\gamma(\mathbf{x}), \mathbf{f_g}]$ to generate RGB values $\mathbf{c}$.

The major challenge lies in the aggregation of noisy semantic features from 2D observations. Unlike color and geometry which remain consistent across views, semantic features vary due to the inherent aleatoric and epistemic uncertainty. We view the learned embedding of color regression as a local regularizer that shares similar decision boundaries with semantics, where sparse but accurate supervision can propagate to nearby regions. This insight aligns with past few research [38, 39], indicating that shared features among geometry, color, and semantics accelerate convergence and assure accuracy with a unified representation. Nonetheless, color alone is insufficient for semantic differentiation, as visually similar instances may belong to different categories. Therefore, we concatenate the positional encoding $\gamma(\mathbf{x})$ with the final-layer features $\mathbf{f_c}$ of the color decoder to provide necessary instance separation, where a single linear layer is learned on-the-fly to regress high-dimensional semantic features from sparse and noisy 2D observations. During optimization, features $\mathbf{f_c}$ derived from the color decoder are detached before being passed to the semantic layer. This ensures that semantic features leverage the shaped feature space without disrupting it, avoiding oscillating gradients introduced by the semantic supervision.

As illustrated in Fig. 3d, the learned color features exhibit patterns aligned with semantic structures shown in Fig. 3a. In addition, the one-blob encoding [21] $\gamma(\mathbf{x})$ shown in Fig. 3b for positional embedding preserves local smoothness geometrically due to the Gaussian kernel activation across adjacent bins. This aligns well with our previous intuition and the design, where the shared positional and color embeddings lead to a nice landscape for semantic decoding. As shown in the experiment section, the strategy not only leads to faster convergence of the semantic regression, but also enhance the accuracy of the final semantic reconstruction. In addition, to balance between the sparse supervision of detected instances and a large portion of unsupervised background areas, we maintain a linear layer to predict uncertainty given the one-blob encoded features $\gamma(\mathbf{x})$. The architecture of the neural representation is provided in the supplementary material for better clarification.

Furthermore, since the neural field is supervised only with task-oriented semantics, it is important to identify background regions lacking supervision, which may otherwise introduce misleading semantics. At the same time, to fully leverage the multi-scale features from the feature pyramid (Sec. 4.1), which are routed to different detection heads for logits based on corresponding scales, we must determine the appropriate scale for each predicted per-point feature at query time, thereby selecting the correct detection head for querying. To address this, we introduce a feature scale head that takes in $\varsigma(\mathbf{x})$ and classifies each point into its most appropriate scale while also identifying points that lack reliable supervision, guiding each point to the correct semantic decoding path.

4.3 Selective Experience Replay for Grounding

To optimize the neural field on-the-fly as indicated in Eq. 2, we adopt the differentiable rendering that bridges the information within the neural field and the observed visual data in the image domain. Given a camera pose $\mathbf{x}$, which determines the camera origin $\mathbf{o}$ and the viewing direction $\mathbf{r}$, a ray is parameterized as $\mathbf{r}(d) = \mathbf{o} + d\mathbf{r}$, where d represents depth along the ray. Volumetric rendering [34, 30] can then be conducted by accumulating per-point contributions along the ray. The rendered outputs for different properties $\hat{\mathbf{y}}_{\mathrm{p}}$, including color $\hat{\mathbf{c}}$, depth $\hat{d}$, semantic feature $\hat{\mathbf{f}}$, and the corresponding feature scale label l are computed as:

$$\hat{\mathbf{y}}_{\mathrm{p}}(\boldsymbol{\theta}) = \frac{1}{\sum_i w_i} \sum_i w_i \mathrm{p}(\mathbf{x}_i; \boldsymbol{\theta}^t), \quad \mathrm{p} \in \{\hat{\mathbf{c}}, s, \hat{\mathbf{f}}, l\}, \tag{4}$$

where $\mathrm{p}(\mathbf{x}_i; \boldsymbol{\theta}^t)$ are the predicted properties at sample $\mathbf{x}_i$ along the ray given model parameters $\boldsymbol{\theta}^t$. The rendering weight w_i is derived from the predicted SDF $s(\mathbf{x}_i; \boldsymbol{\theta}^t)$ at sample $\mathbf{x}_i$ with a truncation distance threshold tr $w_i = \sigma\left(\frac{s(\mathbf{x}_i;\boldsymbol{\theta}^t)}{tr}\right) \sigma\left(\frac{-s(\mathbf{x}_i;\boldsymbol{\theta}^t)}{tr}\right)$.

The model parameters $\boldsymbol{\theta}^t$ are then optimized by minimizing the discrepancy between the rendered outputs and the observations, ensuring that the learned representation aligns with the observed data. Sparse samples are maintained on the fly as the replay buffer to store colored point cloud $\mathbf{c}$, semantic features $\mathbf{f}$, confidence scores c_τ, and binary values u indicating if the corresponding pixels have semantic supervision. The loss function for supervising color and geometry follows the design of CO-SLAM [30] and consists of several components: $\mathcal{L}_2$ losses for color and depth rendering, approximate SDF and feature smoothness losses for improved geometry reconstruction, a free-space loss to enforce spatial consistency, and an additional regularization term for enhanced smoothness. In contrast to the dense observations of color and geometry, the semantic supervision is usually sparse and unbalanced. We separate the sampling of color-geometry rays and semantic rays, where the semantic supervision undergoes selective experience replay that prioritizes representative and valid samples from the buffer for task-oriented optimization. For semantic loss $\mathcal{L}_{sem}$, we employ a $\mathcal{L}_2$ loss between the rendered semantic features and observed semantic features weighted by the task relevance:

Table 1: Comparisons against relevant methods regarding the Top-K accuracy (↑).

Method	top_1	top_5	top_{10}	top_{20}	top_{50}
CLIO [19]	9.63	16.98	20.53	24.56	30.12
ConceptFusion [10]	19.06	35.78	40.91	47.15	54.95
HOV-SG [31]	26.62	52.62	62.23	68.48	77.28
OpenFusion [33]	38.90	57.05	62.75	67.27	75.76
ProGround	**43.26**	**61.65**	**67.69**	**73.66**	**81.50**

Table 2: Top-K accuracy (↑) with different weighting and filtering strategies.

Weighting	Filtering	top_1	top_5	top_{10}	top_{20}	top_{50}
✗	✗	27.01	45.84	56.29	67.05	79.61
✓	✗	39.28	59.63	66.36	72.80	**81.87**
✗	✓	40.14	60.17	66.04	71.59	79.62
✓	✓	**43.26**	**61.65**	**67.69**	**73.66**	81.50

$$\mathcal{L}_{\text{sem}} = c_{\mathcal{T}} \left\|\hat{\mathbf{y}}_{\mathbf{f}}(\boldsymbol{\theta}^t) - \mathbf{f}\right\|_2 . \tag{5}$$

For feature scale supervision, the task can be viewed as an imbalanced multi-class classification problem, where unsupervised regions dominate and certain feature scales may be overrepresented. To address this, we adopt the Focal Loss [17], which down-weights easy examples and focuses training on hard, informative ones:

$$\mathcal{L}_{\text{scale}} = -\sum_{k=1}^{K} \alpha_k (1 - \hat{p}_k)^{\gamma}\, l_k \log \hat{p}_k \tag{6}$$

Here, K is the number of classes (i.e., scale levels plus one for "no supervision"), $l_k \in \{0, 1\}$ is the one-hot ground-truth label for class k, and $\hat{p}_k \in [0, 1]$ is the predicted probability for class k. The term α_k balances class frequencies, and γ controls the down-weighting strength of well-classified examples.

5 Experiments

The experiments aim to evaluate the effectiveness of our method across key aspects regarding scene understanding. To demonstrate its semantic grounding capabilities, we assess localization performance and memory consumption in HM3DSem by comparing against state-of-the-art baselines. Detailed ablation studies are carried out to demonstrate the impact of different feature types, feature-sharing strategies, and feature reshaping.

5.1 Open-Vocabulary Object Localization

Experimental Setup: The experiments are conducted on a desktop PC with an Intel Core i9-12900K CPU and an NVIDIA RTX 3090 GPU. We follow HOV-SG [31] to select eight scenes in the HM3DSem dataset [32], and evaluate with NYUv2 labels[26] containing 894 raw categories. For each scene, we infer the labels of vertexes and compare with ground-truth labels.

Evaluation Details: We use **Top-K Accuracy (TopK-Acc)** for evaluating the accuracy, which measures how accurately the predicted semantic labels match the ground truth within a given region. For each vertex, we first infer its feature scale classification logits and semantic feature vector. This semantic feature along with all candidate category text features are then passed through each scale-specific detection head (i.e., YOLO-World's text-contrastive heads) to compute scale-wise classification logits. Finally, the outputs from all heads are weighted by the predicted feature scale logits to produce the final semantic prediction. If the ground-truth label appears within the top-K

Table 3: Effect of irrelevant prompts on the Top-K Accuracy (↑)

Prompt Ratio	top_1	top_5	top_{10}	top_{20}	top_{50}
Task Only	**43.26**	**61.65**	67.69	73.66	81.50
+ 20%	42.36	60.91	**67.87**	**74.76**	**83.48**
+ 50%	40.09	60.06	66.98	74.27	83.25
+ 100%	38.99	57.86	64.89	73.01	82.93

predicted labels, the prediction is considered correct. Top-K Accuracy is calculated as the ratio of correctly labeled vertices to the total number of vertices.

Results: We select the following baselines: CLIO [19] introduces an *online task-driven* open-set 3D scene graph representation with information bottleneck principle. ConceptFusion [10] constructs a dense point cloud-based map from pixel-aligned features. HOV-SG [31] proposes an *offline* hierarchical open-vocabulary 3D scene graph framework that organizes scene information into floor, room, and object concepts. OpenFusion [33] is a voxel-based *online* method that grounds semantic features through Truncated Signed Distance Function (TSDF) fusion.

As shown in Table 1, our method surpasses all baselines across all Top-K settings, demonstrating superior vertex-wise localization accuracy. Notably, it achieves a +4.36% improvement in Top-1 accuracy over the previous state-of-the-art OpenFusion, and exceeds offline mapping method HOV-SG. The baseline methods rely on hand-crafted cross-view associations that struggle with different segmentation granularities. This results in fragmented instances with noisy semantics, degrading vertex-wise accuracy. Besides, CLIO, ConceptFusion and HOV-SG follow the segment-then-encode fashion for semantic feature extraction, which lacks hierarchical context information as mentioned earlier. CLIO, in particular, underperforms as it associates segments only within a limited temporal window, resulting in more fragments that lack robust feature aggregation, further reducing semantic accuracy.

5.2 Ablation Studies

Influence of Task-relevant Scoring: Semantic weights reflect the task relevance and reliability of samples. To evaluate the effectiveness of the weighting and filtering mechanisms in improving semantic supervision quality, we conduct an ablation study, with results presented in Table 2. The baseline setup uses YOLO-World model features but excludes both confidence weighting and filtering. We then introduce confidence weighting, followed by confidence filtering with a threshold of 0.4.

The results demonstrate a significant performance boost when applying these mechanisms. Confidence weighting alone raises Top-1 accuracy from 27.01% to 39.28%, while semantic filtering further improves it to 40.14%. This indicates that noisy, irrelevant features significantly degrade supervision quality, and reweighting reliable detections or filtering out outliers can yield meaningful gains. Combining both weighting and filtering achieves the best performance across Top-1 to Top-20, suggesting that the prompted filtering well induces task relevance and effectively enhances the semantic grounding accuracy.

Effect of Irrelevant Prompts: To evaluate the impact of task relevancy in prompts, we conduct an ablation study by introducing varying proportions of irrelevant labels from NYUv2 into the prompt list. The baseline setting consists exclusively of task-relevant labels, while the other settings incrementally introduce a certain percentage of irrelevant labels in addition to the task labels. The results are presented in Table 3.

The results show a clear decline in Top-1 and Top-5 accuracy as more irrelevant labels are added, with the most pronounced drop in Top-1. This suggests that extraneous prompts introduce semantic noise, reducing the model confidence in top-ranked predictions, though the impact diminishes at higher K values. These results suggest that incorporating extraneous labels into the prompt list increases ambiguity, making it harder for the model to associate scene features with the correct semantic categories. At the same time, this may also lead the system to detect and retain more instances, potentially improving performance under relaxed ranking criteria (e.g., Top-10 to Top-50).

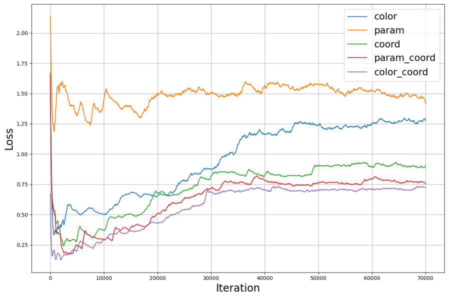


Figure 4: **Semantic loss curves for different sharing strategies.** The fusion of coordinate, parametric, and color features accelerates convergence, with color-coordinate fusion achieving the lowest loss.

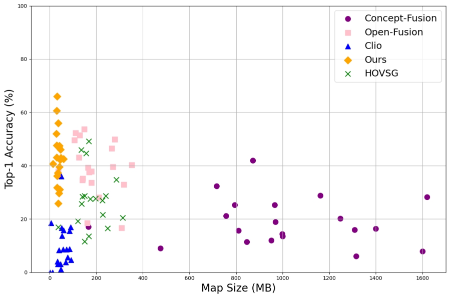


Figure 5: **Map size versus Top-1 Accuracy.** ProGround achieves best performance with minimal memory consumption. Its representation size remains relatively independent of scene scale, as it does not explicitly store features.

Table 4: Top-K accuracies given different choices of the embedding space for semantic decoding.

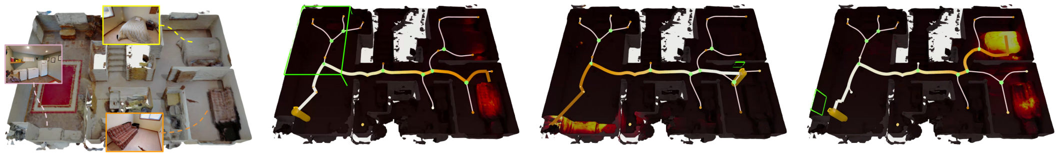
Settings	w/o Coord					w/ Coord				
	top_1	top_5	top_{10}	top_{20}	top_{50}	top_1	top_5	top_{10}	top_{20}	top_{50}
Color	30.25	47.72	54.73	62.00	72.17	**43.26**	**61.66**	**67.70**	**73.67**	**81.50**
Geo	23.41	40.11	46.64	54.22	65.74	40.40	59.56	65.67	71.58	79.37
Param	18.12	30.51	36.31	42.99	54.14	40.63	60.43	66.88	73.00	81.20
Coord	–	–	–	–	–	38.47	56.95	63.13	69.03	78.01

Impact of Feature Space Sharing: Different hidden features influence the optimal performance that semantic learning can achieve. As shown in the loss curve plot in Fig. 4, the hierarchical fusion of color and coordinate features results in the lowest converged loss, indicating that this setup provides the most effective hidden feature space for learning semantics. Solely relying on optimizable features which will be affected by downstream tasks (i.e., color features and sparse parametric embedding) underperforms without positional encoding (i.e., coordinate features), because positional features provide necessary spatial cues to set apart instances. However, task-specific features are not entirely ineffective. Hierarchically fusing deep features with coordinate features further reduces semantic loss. This suggests that while deep features alone are insufficient for semantic learning, they provide valuable cues that enhance performance when combined with more generalizable features from shallow layers. We also carry out the Top-K accuracy analysis across different feature-sharing settings, as shown in Table 4 and obtain numerical results consistent with loss curve analysis.

Analysis of Memory Efficiency: We further analyze the memory consumption of the constructed representations. Fig. 5 presents a scatter plot where each point represents the map size versus Top-1 accuracy for a single scene across different methods, illustrating the trade-off between accuracy and memory usage. Our method achieves significantly higher accuracy while maintaining a compact map representation and remains less sensitive to scene size by avoiding explicit storage of semantic features. Concept-Fusion [10], which stores dense per-point visual features, results in large map sizes exceeding 1GB in many cases, while achieving lower accuracy. HOVSG [31], although more memory-efficient with graph structures, still suffers from explicit feature storage. Clio (blue triangles) maintains a small memory footprint but sacrifices accuracy due to incomplete mesh reconstruction.

5.3 ProGround for Robotic Navigation

Given goals specified in different modalities, we use CLIP encoders to extract features of the target goals for text or image queries, as they are aligned in the same embedding space. Goal features and vertex embeddings are passed through the YOLO-World text-contrastive head to generate class-specific logits, which are then used to cluster vertices for precise instance-level information. Given the queried goal locations, we extract a Voronoi graph from the dense field following [13] for fair-safe



(a) Close-up of goals

(b) Goal: sofa

(c) Goal: fridge

(d) Goal: bed

Figure 6: **Navigation with the neural map.** Given task-oriented semantics from ProGround, the agent navigates toward goals with a path planned by Voronoi graph.

path planning. The robotic navigation is then carried out given the pre-built map, with qualitative results shown in Fig. 6.

6 Conclusions

In this work, we introduced ProGround, a prompt-then-ground paradigm for task-aware online neural implicit mapping. By prioritizing task-relevant, high-confidence features, our method mitigates semantic noise and ambiguity while maintaining a compact and queryable scene representation. Unlike traditional segment-then-encode approaches, ProGround refines the data distribution into a highly task-oriented one, ensuring more reliable feature grounding.

Our approach achieves state-of-the-art vertex-wise semantic accuracy in the Habitat-Sem[32] dataset, even surpassing some offline methods, while maintaining memory efficiency through implicit encoding. Extensive ablations showcase the impact of introducing task-awareness into mapping, validate our architectural design, highlighting the effectiveness of feature sharing for online convergence. We also demonstrate the applicability of ProGround to robotic navigation.

Acknowledgements

This work is supported by Xiaomi Foundation and Wuxi Research Institute of Applied Technologies, Tsinghua University (20242001120).

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Intl. Conf. on Computer Vision (ICCV)*, pages 9650–9660, 2021.

[2] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[3] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874*, 2022.

[4] 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.

[5] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10995–11005, 2023.

[6] 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 *Intl. Conf. on Learning Representations (ICLR)*, 2024.

[7] 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*, pages 540–557. Springer, 2022.

[8] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2023.

[9] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, London, UK, 2023.

[10] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. In *Robotics: Science and Systems (RSS)*, 2023.

[11] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Intl. Conf. on Computer Vision (ICCV)*, pages 19729–19739, 2023.

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Intl. Conf. on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[13] Zijia Kuang, Zike Yan, Hao Zhao, Guyue Zhou, and Hongbin Zha. Active neural mapping at scale. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7152–7159. IEEE, 2024.

[14] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[15] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[16] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[18] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[19] Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo, Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *arXiv preprint arXiv:2404.13696*, 2024.

[20] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

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

[22] Ri-Zhao Qiu, Yafei Hu, Ge Yang, Yuchen Song, Yang Fu, Jianglong Ye, Jiteng Mu, Ruihan Yang, Nikolay Atanasov, Sebastian Scherer, and Xiaolong Wang. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Intl. Conf. on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[24] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv: Arxiv-2210.05663*, 2022.

[25] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *Conf. on Robot Learning*, pages 405–424. PMLR, 2023.

[26] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

[27] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Intl. Conf. on Computer Vision (ICCV)*, pages 6229–6238, 2021.

[28] Muer Tie, Julong Wei, Ke Wu, Zhengjun Wang, Shanshuai Yuan, Kaizhao Zhang, Jie Jia, Jieru Zhao, Zhongxue Gan, and Wenchao Ding. O2v-mapping: Online open-vocabulary mapping with neural implicit representation. In *European Conference on Computer Vision*, pages 318–333. Springer, 2025.

[29] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[30] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023.

[31] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *Robotics: Science and Systems (RSS)*, 2024.

[32] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4927–4936, 2023.

[33] Kashu Yamazaki, Taisei Hanyu, Khoa Vo, Thang Pham, Minh Tran, Gianfranco Doretto, Anh Nguyen, and Ngan Le. Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9417. IEEE, 2024.

[34] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[35] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conf. on Robot Learning*, pages 284–301. PMLR, 2023.

[36] Albert J Zhai and Shenlong Wang. Peanut: predicting and navigating to unseen targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10926–10935, 2023.

[37] 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.

[38] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *Intl. Conf. on Computer Vision (ICCV)*, 2021.

[39] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and Andrew J Davison. ilabel: Interactive neural scene labelling. *arXiv preprint arXiv:2111.14637*, 2021.

[40] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16793–16803, 2022.

[41] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024.