# Topo-Field: Topometric Mapping With Brain-Inspired Hierarchical Layout-Object-Position Fields

Jiawei Hou, Wenhao Guan, Longfei Liang, Jianfeng Feng, Xiangyang Xue, *Member, IEEE*, and Taiping Zeng

***Abstract*—Mobile robots require comprehensive scene understanding to operate effectively in diverse environments, enriched with contextual information such as layouts, objects, and their relationships. Although advances like neural radiance fields (NeRFs) offer high-fidelity 3D reconstructions, they are computationally intensive and often lack efficient representations of traversable spaces essential for planning and navigation. In contrast, topological maps are computationally efficient but lack the semantic richness necessary for a more complete understanding of the environment. Inspired by a population code in the postrhinal cortex (POR) strongly tuned to spatial layouts over scene content rapidly forming a high-level cognitive map, this work introduces Topo-Field, a framework that integrates Layout-Object-Position (LOP) associations into a neural field and constructs a topometric map from this learned representation. LOP associations are modeled by explicitly encoding object and layout information, while a Large Foundation Model (LFM) technique allows for efficient training without extensive annotations. The topometric map is then constructed by querying the learned neural representation, offering both semantic richness and computational efficiency. Empirical evaluations in multi-room environments demonstrate the effectiveness of Topo-Field in tasks such as position attribute inference, query localization, and topometric planning, successfully bridging the gap between high-fidelity scene understanding and efficient robotic navigation.**

***Index Terms*—Bioinspired robot learning, cognitive map, mapping, neural field, representation Learning.**

Received 10 December 2024; accepted 1 April 2025. Date of publication 10 April 2025; date of current version 21 April 2025. This article was recommended for publication by Associate Editor K. Skinner and Editor J. Civera upon evaluation of the reviewers' comments. The work of Taiping Zeng was supported by Fudan startup funding. This work was supported by the Shanghai Technology Development and Entrepreneurship Platform for Neuromorphic and AI SoC. *(Corresponding author: Taiping Zeng.)*

Jiawei Hou, Wenhao Guan, and Xiangyang Xue are with the School of Computer Science, Fudan University, Shanghai 200437, China (e-mail: jwhou23@m.fudan.edu.cn; whguan21@m.fudan.edu.cn; xyxue@fudan.edu.cn).

Longfei Liang is with the Shanghai Neuhelium Neuromorphic Intelligence Technical Company Ltd., Shanghai 200090, China (e-mail: longfei.liang@neuhelium.com).

Jianfeng Feng and Taiping Zeng are with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200437, China (e-mail: jffeng@fudan.edu.cn; zengtaiping@fudan.edu.cn).

The open-source code is available at: https://github.com/fudan-birlab/Topo-Field.

Digital Object Identifier 10.1109/LRA.2025.3559836

## I. Introduction

MOBILE robots are rapidly moving from research labs to widespread use. For these robots to operate autonomously in complex environments, a deep understanding of their surroundings is crucial [1]. Hierarchical graph-like scene representation along with detailed environmental reconstruction enabling efficient path planning, will be key for robotic deployment in real-world scenarios [2]. This necessity arises from a fundamental dual challenge in robotics: achieving accurate geometric reconstruction to facilitate precise local obstacle avoidance and semantic interpretation, while simultaneously leveraging hierarchical abstraction to enable computationally efficient global planning and navigation.

Recently, detailed environmental reconstruction has made great progress in producing lifelike 3D reconstructions [3], [4], [5], in which NeRF [6] is a prime instance. As improvements, works like [7], [8] introduce semantic information for better scene understanding. Further, features powered by Large-Foundation-Models (LFMs), trained on massive datasets across various scenes, are employed with general knowledge for open scene understanding [9], [10], [11], [12], [13], [14]. However, it is computationally demanding and lacks layout information using detailed neural fields for planning and navigation.

In contrast, existing topological maps for path planning and navigation in complex environments are often derived from LiDAR Simultaneous-Localization-and-Mapping (SLAM) using 3D dense submaps [15] or visual SLAM by clustering free-space regions and extracting occupancy information from point clouds [2]. While this approach increases path planning accuracy, computing topology with traditional methods comes with high computational costs and tends to strip away essential semantic information, reducing the robot's ability to fully understand the environment, which is critical for advanced autonomous functions such as language/image-prompted localization and navigation.

To this end, we propose to build a neural representation as spatial knowledge and construct a topometric map based on this, originating from a brain-inspired approach. Theoretically, neuroscientists have long discovered that animals process their surroundings using topological coding, forming what is known as a "cognitive map" [16], a concept embodied by place cells (hippocampal neurons that activate in specific places to form place fields) [17]. These place cells, along with spatial view cells (neurons selectively responsive to different spatial perspectives, aiding in the integration of visual and spatial information) [18], respond to specific scene contents. More recently, research has

shown that a population code in the postrhinal cortex (POR, a brain region serves as a key input area to the hippocampal system) is strongly tuned to spatial layout rather than scene content [19], capturing spatial representations relative to environmental centers to form a high-level cognitive map from egocentric perception to allocentric understanding [20].

Most related works either do not explicitly represent the layout features [21] or build the topo-map in a clustering and incremental mapping way [22], [23]. On the contrary, we intuitively abstract the neural representations of space to build topo-field in three key aspects: 1) The cognitive map corresponds to a topometric map, which uses graph-like representations to encode relationships among its components, e.g. layouts and objects. 2) The population of place cells is analogous to a neural implicit representation with position encoding, enabling location-specific responses. 3) POR, which prioritizes spatial layouts over content, aligns with our spatial layout encoding of connected regions.

This work proposes a Topo-Field, integrating the Layout-Object-Position (LOP) association into neural field training and constructing a topometric map based on the learned neural implicit representation for hierarchical robotic scene understanding. By inputting RGB-D sequences, objects and background contexts are encoded separately as contents and layout information to train a neural field, forming a detailed scene representation. A contrast loss against features from LFMs is employed, resulting in little need for annotation. Further, a topometric map is built by querying the learned field, which is efficient for navigable path planning. To validate the effectiveness of Topo-Field, we conduct quantitative and qualitative experiments on several multi-room apartment scenes evaluating the abilities including position attributes inference, text/image query localization, and planning.

Our contributions can be listed as follows:

- *Brain-inspired Topo-Field:* We introduce a Topo-Field that combines neural scene representation with efficient topometric mapping, enabling hierarchical robotic scene understanding and navigable path planning.
- *Cognitive Map Representation:* Inspired by the population code in postrhinal cortex (POR) strongly tuned to spatial layouts over scene content rapidly forming a high-level cognitive map, we incorporate the concepts of neural representations of spatial layouts, objects, and place cells to construct hierarchical robotic topo-maps.
- *Layout-Object-Position (LOP) Representation:* We develop an implicit neural representation associating layout, object, and position information, which is explicitly supervised using an LFM-powered strategy, requiring minimal human annotation.
- *Topometric Map Construction:* We propose a two-stage pipeline for building a topometric map by querying the learned neural field and validating edges among vertices using LLMs, enabling efficient path planning.

## II. Related Works

### A. Dense Representation With Neural Radiance Field

Detailed 3D scene reconstruction has made great efforts in producing lifelike results, among which NeRF (Neural Radiance Fields) [6] has widely attracted researchers' attention. A popular research direction is to integrate semantics with NeRF to achieve a more comprehensive understanding of scenes [7], [8]. Recently, several robotic works have demonstrated that features from LFMs can be used for self-supervised learning, which reduces the costly manual annotation [9], [10], [11], [12], [13], [14]. However, they focus on object semantics but do not include layout-level features. RegionPLC [21] considered region information but with no explicit representation of layout features. In contrast, in our work, CLIP [24] and S-BERT [25] are employed to generate vision-language and semantic features for objects and layout respectively.

### B. Topometric Map for Scene Structure Understanding

Using detailed neural fields for planning and navigation is computationally demanding, on the other hand, hybrid topometric mapping has been known for its efficiency in terms of managing the information and being queried for downstream tasks [26], [27]. However, most topological maps have not introduced information such as semantics. Concept-graph [28] made a step forward utilizing LFM to model the object structure with topology. SceneGraphFusion [29] created a globally consistent scene graph by incrementally fusing predictions of a graph neural network (GNN). However, they focused on the scene contents at object level and neither of them included layout-level information. CLIO [22] built a task-driven scene graph forming task-relevant clusters of primitives. HOV-SG [23] proposed using feature point cloud clustering and mapping in an incremental approach. Unlike the incremental mapping and clustering-based method, We query the learned representation of object and region separately with fewer vertices representing the same scene. Each vertice clearly represents only one object or region with its attributes. Hydra [30] realized impressive real-time 3D scene graph reconstruction, however, it faced challenges labeling some of the topology vertices and relationship of edges because of the costly and close-set segmentation. On the contrary, we leverage open-set vision-language and semantic encoders for feature extraction and employ LLM to help validate the edge relationship.

### C. Spatial Understanding With Layout Information

Generally, topology is built based on clustering from occupancy information or Voronoi diagrams [31], regardless of the contents and layout relationship. However, neuroscience findings suggest a mechanism to form a high-level cognitive map from egocentric perception to allocentric representation [16], [20]. Place cells [17], as the embodiment of cognitive map, together with spatial view cells show activity to contents [18]. Recently, Patrick et al. [19] showed that a population code in the POR is more strongly tuned to the spatial layout than to the content in a scene. This suggests that there are specialized cells and signaling mechanisms to process layout in the process of scene understanding, which captures the spatial layout of complex environments to rapidly form a high-level cognitive map representation [20]. Inspired by the above research, we mimic the neural scene understanding mechanism by employing egocentric neural field with content and layout knowledge to construct allocentric topometric map.

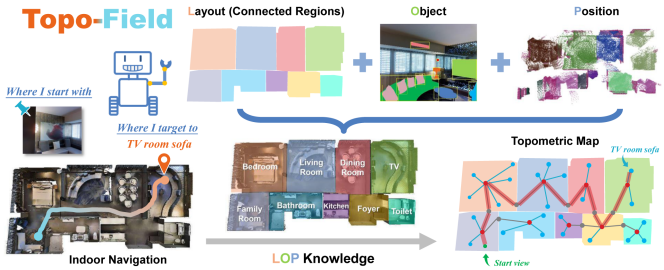


Fig. 1. Illustration of the Topo-Field strategy and capabilities. Hierarchically dividing scene information into layout, object, and position to model them explicitly, layout-object-position associated knowledge enables robots with a topometric map representing the scene and planning navigable path to realize a more comprehensive spatial cognition.

## III. Overview

This work proposes a framework that integrates Layout-Object-Position (LOP) associations into a neural field and constructs a topometric map from this learned representation. The overview strategy is shown in Fig. 1. To achieve this, first, we need to train a scene-dependent implicit function, denoted as

$$F : \mathbb{R}^3 \rightarrow \mathbb{R}^n, \tag{1}$$

where for any 3D point $P$ in space, $F(P)$ is supposed to be its corresponding embeddings $\mathcal{E}\{(e_v, e_s)\} \in \mathbb{R}^n$. $\mathcal{E}$ consists of vision-language embedding $e_v$ and semantic embedding $e_s$. We constrain $(e_v, e_s)$ to match with embedding space of pre-trained vision-language and language models. The target embeddings are encoded from observed images of the scene and relationships of 3D points and embeddings are constructed based on back-projecting pixels with corresponding depth and camera poses. Target feature processing and training strategy are described in Sections IV-A and IV-D. Learned field applications are discussed in Section IV-C1.

Based on the trained $F$ and a region set of the scene $\{r_1, r_2, \ldots, r_m\}$, we sample 3D points $p$ from the scene, calculate and compare cosine similarity of $F(p)$ with embeddings of language-encoder-processed $\{r_1, r_2, \ldots, r_m\}$ to get region layout distribution of the scene. Points that belong to the same regions are clustered and form a region vertex set. Objects in images perceived in the previous process to train $F$ form the object vertices. Their relationship is inferred with the help of LLM. In this way, a topometric map is built denoted as

$$G = (V, E), \tag{2}$$

where vertices $V$ include object vertices $\mathbf{v}_o$ and region vertices $\mathbf{v}_r$ and edges $E$ include edges between objects $\mathbf{e}_{o-o}$, edges between regions $\mathbf{e}_{r-r}$, and edges between object and region $\mathbf{e}_{o-r}$. The topo-map architecture and construction pipeline are described in Section IV-C2.

## IV. Method

### A. Target Feature Processing

RGB-D image sequences with poses are accepted as input for training $F$. For RGB image sequences, depth point clouds and camera poses can also be estimated through COLMAP [32] or simultaneous localization and mapping (SLAM). The only employed GT annotation is the layout distribution of environment. The region of each 3D point $P$ is denoted as $r_P \in R = \{r_1, r_2, \ldots, r_q\}$, where $q$ is the number of regions. Such information is available in datasets like Matterport3D [33], whereas partitioning the buildings needs little human labor. In most human-made buildings spatial, layouts are easily available divided by straight walls. Simply drawing lines from top-down view according to walls can form a rule to bound 3D points to different regions.

For object pixels embedding $e_{p_o}$ processing, the pipeline follows CLIP-Fields [9] employing CLIP [24] $C$ and Sentence-BERT [25] $S$ as encoders. The difference is the semantic label of objects is prompted in the form of "$l$ *in* $r$", where $l$ is the object label and $r$ is the region label. What's more, the background appearance is also considered which we proposed to include context information for region layout. For background pixels $p_b$, per-pixel feature of image $I$ is encoded. Its related region $r_{p_b} \in R$ is regarded as the text label and embedding of $p_b$ can be denoted as $e_{p_b} = \{C(I), S(r_{p_b})\}$.

Then, pixel-wise embeddings are back-projected to 3D space based on depth and pose and form a distilled 3D feature point cloud. Consequently, the target feature space $\mathcal{E}\{(e_v, e_s)\}$ consists of object and layout features, where $(e_v, e_s)$ directs from $\{e_{p_o}, e_{p_b}\}_{p_o, p_b \in P}$. The pipeline is shown in Fig. 2.

### B. Scene Neural Encoding

Our proposed Topo-Field involves an implicit mapping function to encode the 3D position into a spatial vector representation $g : \mathbb{R}^3 \rightarrow \mathbb{R}^d$ and separate heads $h : \mathbb{R}^d \rightarrow \mathbb{R}^n$ processing encodings to match the target feature space $\mathcal{E}\{(e_v, e_s)\}$. we employ the Multi-scale Hash Encoding (MHE) introduced in Instant-NGP [34] as $g$ with embedding dimension $d = 144$. As for heads, we follow [9] and employ Multi-Layer Perceptron (MLP) network $h_v : \mathbb{R}^d \rightarrow f_v$ for obtaining vision-language features and $h_s : \mathbb{R}^d \rightarrow f_s$ for semantic features. The model is shown in Fig. 2.

In this way, given a posed RGB-D image, the target feature of each pixel is processed as mentioned in Section IV-A denoted as $\mathcal{E}\{(e_v, e_s)\}$. At the same time the related pixel in depth image is back-projected into 3D space according to depth and pose value and processed by the above mentioned $g, h$ to form $(f_v, f_s)$. A contrastive loss is conducted between $(e_v, e_s)$ and $(f_v, f_s)$ to train the neural representation. Training details are declared in Section IV-D.

### C. Topometric Mapping

With the function and feature representation mentioned above, we can integrate 3D positions with the object and region information and construct a topometric map. The topo map construction process is formed in a mapping and updating strategy, while the implicit neural representation is introduced and queried as scene knowledge in this process. Detailed pipeline is introduced as follows.

*1) Knowledge From Learned Neural Field: Position Attributes Inference:* Using spatial 3D point $P$ as input, assuming a collection of space regions $R$ (e.g., "living room", "bathroom", "bedroom", ...), we compute the vision-language features $\mathcal{C}_R = \{C(r_1), C(r_2), \ldots, C(r_m)\}$ and semantic features $\mathcal{S}_R = \{S(r_1), S(r_2), \ldots, S(r_m)\}$ using CLIP [24] encoder $C$ and Sentence-BERT [25] encoder $S$, where $m$ is the number of rooms. Then the cosine similarity between $F(P) = \{(f_v, f_s)\}_P$ and $\{\mathcal{C}_R, \mathcal{S}_R\}$ is calculated to find the most likely region to

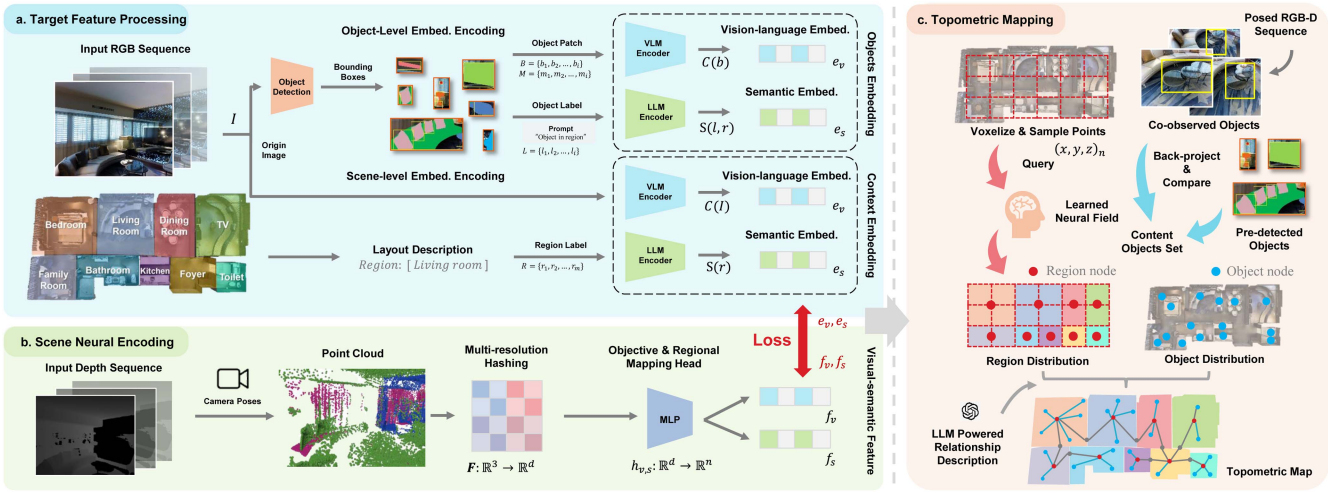


Fig. 2. Pipeline of the Topo-Field. (a) The ground truth generation of layout-object-position vision-language and semantic embeddings for weakly-supervising. (b) The neural implicit network mapping 3D positions to target feature space. A contrastive loss is optimized against each other. (c) Topometric mapping process with trained neural field.

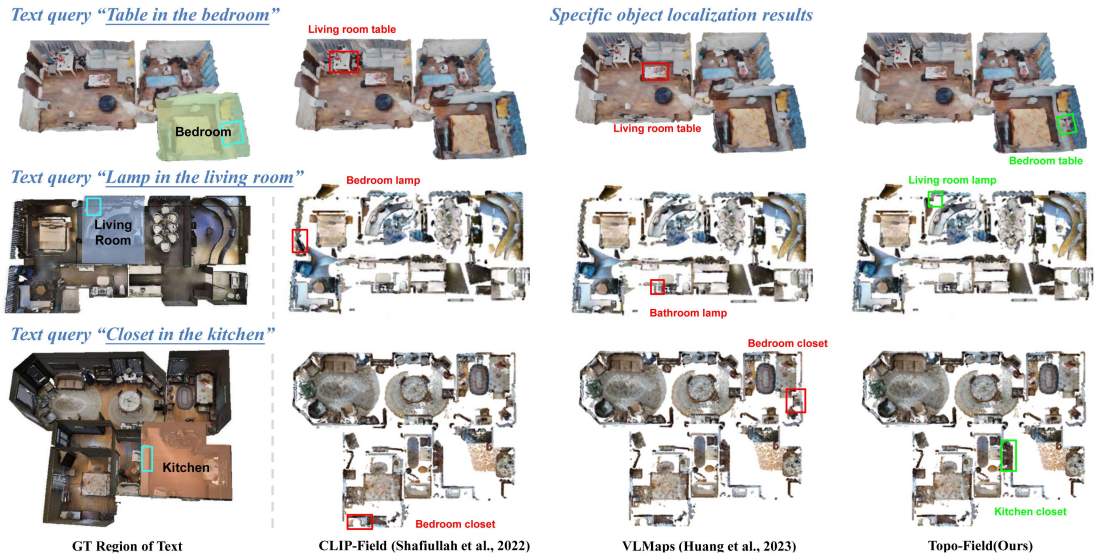


Fig. 3. Qualitative comparison of text query localization results among state-of-the-art methods and our method with text input in the form of "*object in the region*". Blue box shows the ground truth bounding box of object. Red box means miss-predicted box, while green box means the correctly predicted results.

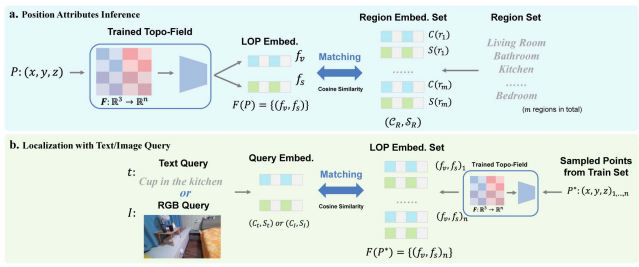


Fig. 4. Capabilities of the learned neural field. (a) The attributes inference using position input. (b) The LOP association helped localization of text and image queries.

which $P$ belongs. The inference process is shown in Fig. 4(a). Similarly, the object information of $P$ can be inferred with the same approach replacing the region set $R$ with object set $O$.

*Localization with Text/Image Query:* For natural language text input $t$ (e.g., "cup in the bedroom"), most existing robotic scene representations struggle to locate specific objects of interest (e.g., differentiating between cups in the living room and the bedroom). However, with our proposed Topo-Field that includes region information, we can calculate the cosine similarity between $\{\mathcal{C}_t, \mathcal{S}_t\}$ and the embeddings $F(P^*) = \{(f_v, f_s)\}_{P^*}$ to find the most likely position of queries, where $P^*$ are sampled from 3D points set to train $F$. As for image input $I$, we can calculate the cosine similarity of $\{\mathcal{C}_I, \mathcal{S}_I\}$ with $F(P^*) = \{(f_v, f_s)\}_{P^*}$ in the same way to find the 3D points set with highest similarity. Localization process of text query and image query is shown in Fig. 4.

*2) Topometric Map Construction:* As defined in Section III, topometric map $G = (V, E)$ consists of vertices and edges. We define a vertex $\mathbf{v}$ : {id, vertex_type, class, bounding_box, caption} and edge $\mathbf{e}$ : {id, edge_type, start_vertex, end_vertex,

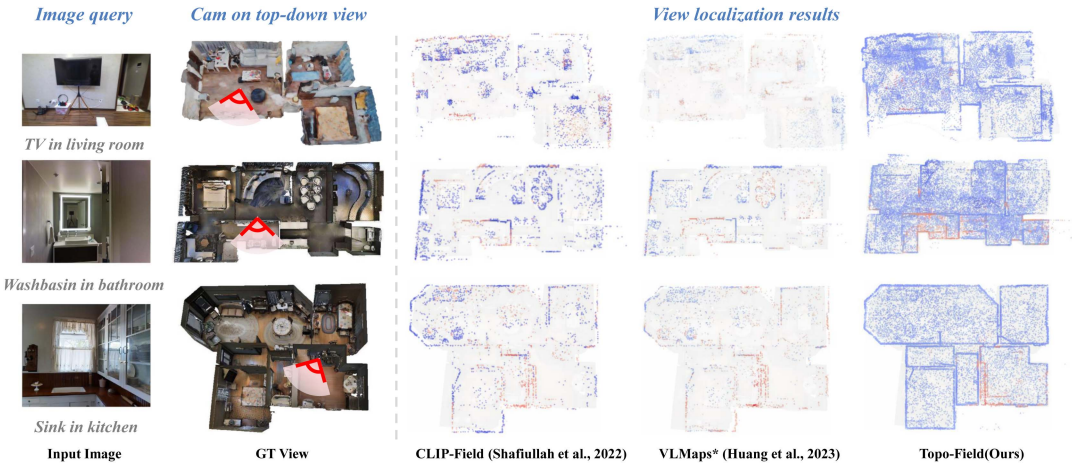


Fig. 5. Qualitative comparison of image query localization results in heatmaps form among state-of-the-art methods and our method with image input. Our approach localizes the position of queried image in an exact smaller range.

relationship, caption}. Inspired by the mental cognitive map representation, such allocentric topo-map is constructed by leveraging the neural field $F$ learned from egocentric observations as scene memory and querying it.

*Mapping:* We first averagely sample $k$ points $P_{1,\dots,k}$ in the environment by dividing the scene into voxel grids of $0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ (for performance-efficiency trade-off according to our experiments) from top-down view and regarding the center point of each voxel as sampled point. The region of each point is inferred according to Section IV-C1. Points $p*$ belonging to the same predicted region $r_j$ is clustered to form the region vertex $\mathbf{v}_{r_j}$ whose {bounding_box} attribute is set according to the upper-bound and lower-bound of $p*$ positions. {class} and {caption} is set according to the region label. For object vertex, inheriting the object detection results by Detic [35] when processing target embeddings illustrated in Section IV-A as [9], objects with high confidence (more than 60%) are recorded as object vertices candidates. Their attributes are set according to the detection results. With the mapped vertices, we leverage LLM to describe the layouts with connectivity, distances, and relationships of regions and objects in JSON format based on the vertices' attributes and poses. During this process, edges are built among vertices and relationships are validated with the help of LLM. For object-object edge $\mathbf{e}_{o-o}$, we follow [28] which mainly considers bounding-box overlap and position relations. For object-region edge $\mathbf{e}_{o-r}$, we consider an object belongs to the region if the object bounding box is in the region bounding box and filter the unreasonable relation noise powered by LLM (e.g., it's almost impossible that a bike is in bedroom). For region relationships, the adjacency and position relationship (e.g., north, south, east, west, ...) of region bounding box is considered. Fig. 2 shows the pipeline of metric-topological map construction.

*Updating:* RGB-D image sequence for training $F$ or a newly captured sequence can be used for constructed map fine-tuning. For object vertices, if an object is detected by more than 3 frames in sequence, the object bounding box will be compared with the constructed vertices. A new vertex will be added if no existing vertex corresponds to it. For region vertices, we calculate embeddings $F(p_I)$ of sampled back-projected pixels $p_I$ in each image $I$. $F(p_I)$ will be matched with the constructed region set $r_{1,\dots,m}$, and extent of a region $r$ will be updated if $F(p_I)$ matches $\{\mathcal{C}_r, \mathcal{S}_r\}$ and $p_I$ exceeds the {bounding_box} extent of vertex $\mathbf{v}_r$. LLM will be called to update edges each 50 frames. Fig. 6 shows an example of the initially mapped region distribution and the updated one.

### D. Training

The pipeline of ground truth data generation is described in Section IV-A to train $F$. To fit the implicit representation introduced in Section IV-B to the target feature space, we design the loss function through a contrastive approach. For the vision-language feature optimization, the tempered similarity matrix on point $P$ is denoted as

$$\mathrm{Sim}_v = \tau\{f_v\}_P\{e_v\}_P, \tag{3}$$

where $\tau$ is the temperature term, $\{f_v\}_P$ and $\{e_v\}_P$ is the calculated implicit representation feature and target embedding according to $P$. Using cross-entropy loss, the vision-language loss can be calculated as

$$\mathcal{L}_v = -\exp(-\mathrm{dist}_P)\left(H\left(\mathrm{Sim}_v\right) + H\left(\mathrm{Sim}_v{}^T\right)\right), \tag{4}$$

where $\mathrm{dist}_P$ is the distance from $P$ to camera, and $H$ is the cross-entropy function. For the semantic loss, similarity on points $P$ can be calculated as

$$\mathrm{Sim}_s = \tau\{f_s\}_P\{e_s\}_P. \tag{5}$$

Similarly, semantic loss can be denoted as

$$\mathcal{L}_s = -\mathrm{conf}\left(H\left(\mathrm{Sim}_s\right) + H\left(\mathrm{Sim}_s{}^T\right)\right), \tag{6}$$

where $conf$ is the prediction confidence from the detection model. The total loss is computed by:

$$\mathcal{L} = \mathcal{L}_v + \mathcal{L}_s. \tag{7}$$

In our experiments, an NVIDIA RTX3090 GPU is utilized and the batch size is set to 12544 to maximize the capability of our VRAM. As model instances, CLIP with SwinB is employed in Detic [35], CLIP [24] encoder is ViT-B/32 and Sentence-BERT [25] encoder is all-mpnet-base-v2. The MHE has 18 levels of grids and the dimension of each grid is 8, with $\log_2$ hash map size of 20 and only 1 hidden MLP layer of size 600. We train the neural implicit network for 100 epochs with optimizer $Adam$, employing a decayed learning rate of $1E-4$ and $3E-3$ decay

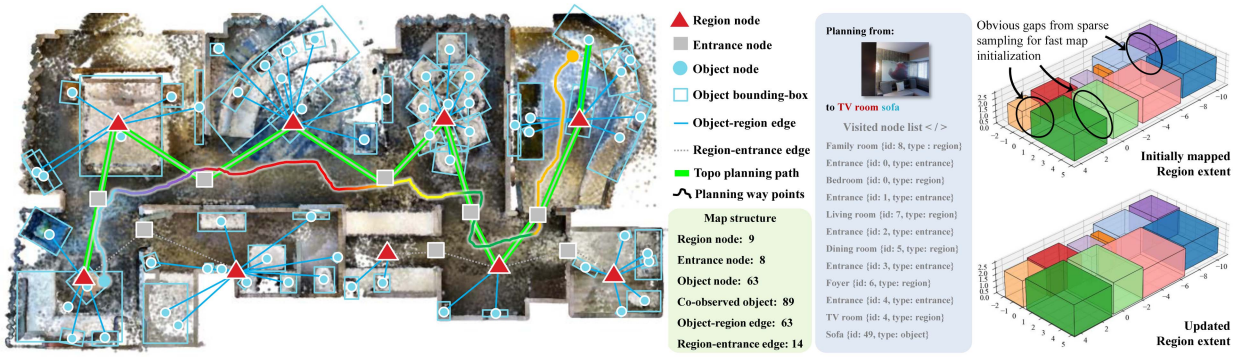


Fig. 6. Topometric map construction example. The topometric map is represented as a graph from a top-down view. Map structure shows number of vertices and edges. A planning path from a seen view to target is shown as an example employing topometric map, the path is highlighted in green showing the related vertices and edges. Visited vertices are listed on the right. The line with gradient colors represents the waypoints based on the planning results while different colors represent different predicted regions of waypoints. On the right, we show the the initially mapped region distribution and the updated one. Obviously the updating process fill the gap caused by sparse point sampling for fast map initialization.

rate. Each epoch contains $3E6$ samples. Codes and scripts are released in supplementary for reproducibility.

## V. Experimental Results

Our experiments are conducted on real-world datasets to validate the established layout-object-position association. The data environment is of single-floor residential buildings with multiple rooms which is the common working scenario of household robots widely studied in this field. We employed Matterport3D [33] as well as apartment environment [36] dataset to demonstrate that our approach can be generalized in diverse scenarios.

### A. Position Attributes Inference

To demonstrate the built LOP association integrates positions with layout features, we designed experiments that accept 3D positions as input to infer the region information. For quantitative evaluation, we divide the RGB-D sequences into training and testing sets randomly in the $8:2$ ratio. RGB-D images in training set are employed to train Topo-Field and compared methods. 3D positions back-projected from the depth-pose tuples in the test set are leveraged as test input. As described in Section IV-C1, for each input 3D position $p$, point-wise feature $F(p)$ is calculated and compared with the embedding $\mathcal{E}_R$ of the given region set $R$, validated by cosine similarity. The region with the highest embedding similarity is considered as prediction result of the position and compared with the GT belonged region. The region prediction accuracy is employed as metric. Table II shows the region inference results on 10 real-world scenes in Matterport 3D [33] with different scales and layouts indicating the average accuracy of Topo-Field exceeds 85%.

*Baselines:* The same training and testing set is leveraged among our method and baselines. For neural-field-based methods (CLIP-Fields [9], LERF [11], and ours), the training epochs, learning rate, and embedding dimensions are aligned. The vision-language encoder, which is employed to generate the target label embedding, is kept the same as their letter and publicly available code base. For VLMaps [10], the LSeg [37] and CLIP [24] are employed. For RegionPLC [21], we implemented it with the Matterport3D data in the annotation-free open-world manner as described in its letter for fairness. For all baselines, the point-wise embeddings from each scene representation are directly compared to the accordingly encoded $\mathcal{E}_R$ leveraging the encoder mentioned in separate approaches in the cosine similarity manner.

TABLE I
Quantitative Comparison of Text Query Localization Results on Different Scenes From the Matterport3D Dataset

| Methods | Scene1 | | Scene2 | | Scene3 | | Scene4 | |
|---|---|---|---|---|---|---|---|---|
| | Dist. | Acc. | Dist. | Acc. | Dist. | Acc. | Dist. | Acc. |
| CLIP-Field(2022) | 2.97 | 0.24 | 3.35 | 0.21 | 2.98 | 0.20 | 3.06 | 0.17 |
| VLMaps(2023) | 2.78 | 0.28 | 3.63 | 0.16 | 3.05 | 0.24 | 3.12 | 0.12 |
| LERF(2023) | 2.86 | 0.32 | 2.82 | 0.11 | 3.49 | 0.17 | 3.04 | 0.20 |
| HOV-SG(2024) | 1.24 | 0.76 | 1.38 | 0.79 | 0.82 | 0.81 | 0.92 | 0.84 |
| Topo-Field(Ours) | **0.92** | **0.85** | **0.86** | **0.84** | **0.36** | **0.95** | **0.27** | **0.97** |

The average distance (m) from targets to localized point clouds and the accuracy evaluating whether predicted positions are in correct regions are used as metrics.

### B. Localization With Prompt Queries

*Localization with Text Queries:* For objects of the same category in different regions, we input the linguistic text query in the form of "object in the region" and infer the specific location of the target, comparing the results with the predictions from current state-of-the-art vision-language algorithms. Fig. 3 and Table I show qualitative and quantitative results on different scenes demonstrating our advantage in accuracy and distance from targets. The average distance $(m)$ of predicted point cloud and ground truth point cloud is evaluated, together with counting whether the center of predicted points is in the correct region. Ground truth comes from the Matterport3D-provided object instance labels. As the results show, topology indeed helps layout-aware problems, specifically the explicit representation of layout information.

*Localization with Image Queries:* To validate the help of region information in the image view localization task, the localization results are shown in Fig. 5 in the form of heatmaps and Table III shows the quantitative results which evaluates the weighted average distance of the target view and localized point cloud among all samples in a scene, using similarity as weight. VLMaps* is a self-implemented version with CLIP [24] encoder, because origin VLMaps [10] does not implement the image localization. Results show that Topo-Field constrains

TABLE II
COMPARISON OF POSITION ATTRIBUTES INFERENCE RESULTS ON THE TEST SET OF DIFFERENT SCENES FROM THE MATTERPORT3D DATASET

| Methods | Scene1 | Scene2 | Scene3 | Scene4 | Scene5 | Scene6 | Scene7 | Scene8 | Scene9 | Scene10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CLIP-Field(2022) | 0.242 | 0.165 | 0.130 | 0.142 | 0.127 | 0.138 | 0.227 | 0.200 | 0.102 | 0.060 |
| VLMaps(2023) | 0.177 | 0.194 | 0.127 | 0.098 | 0.148 | 0.187 | 0.199 | 0.221 | 0.092 | 0.087 |
| LERF(2023) | 0.268 | 0.189 | 0.165 | 0.153 | 0.136 | 0.169 | 0.216 | 0.252 | 0.110 | 0.091 |
| RegionPLC(2023) | 0.290 | 0.202 | 0.173 | 0.168 | 0.152 | 0.154 | 0.243 | 0.248 | 0.086 | 0.088 |
| HOV-SG(2024) | 0.852 | 0.871 | 0.842 | 0.887 | 0.855 | **0.860** | 0.243 | 0.882 | **0.834** | 0.813 |
| Topo-Field(Ours) | **0.886** | **0.900** | **0.884** | **0.894** | **0.872** | 0.858 | **0.901** | **0.897** | 0.821 | **0.839** |
| Position Samples | 169k | 185k | 111k | 112k | 106k | 176k | 130k | 121k | 205k | 211k |

The region prediction accuracy of sampled 3D points is used as metric.

TABLE III
QUANTITATIVE COMPARISON OF IMAGE QUERY LOCALIZATION RESULTS WITH OTHER METHODS

| Methods | Scene1 | Scene2 | Scene3 | Scene4 |
|---|---|---|---|---|
| CLIP-Field(2022) | 2.541 | 2.748 | 2.922 | 2.651 |
| VLMaps*(2023) | 2.112 | 1.894 | 1.181 | 1.595 |
| LERF(2023) | 1.276 | 1.175 | 1.148 | 1.129 |
| AVLMaps(2023) | 1.228 | 1.205 | 0.852 | 0.975 |
| Topo-Field(Ours) | **0.742** | **0.830** | **0.374** | **0.327** |

The similarity weighted average distance (m) between the target view point cloud and the predicted point cloud is evaluated. VLMaps* is a self-implemented version with image localization ability.

TABLE IV
TOPO-GRAPH VERTICES AND EDGES ACCURACY COMPARISON. EVALUATION METRICS FOLLOW THE CG [28]

| Methods | Object vertex | Region vertex | Edge |
|---|---|---|---|
| SGF(2021) | 0.57 | - | 0.86 |
| Hydra(2022) | 0.71 | * | 0.88 |
| CG(2024) | 0.69 | - | 0.92 |
| CLIO(2024) | 0.71 | 0.95 | 0.93 |
| HOV-SG(2024) | 0.72 | 0.91 | 0.96 |
| Topo-Field(Ours) | **0.74** | **1.00** | **0.96** |

*Hydra does not predict region vertex semantics.

the localization results to a smaller range in the exact region. We sampled more than 40 images on each scene from Apartment [36] and Matterport3D [33].

### C. Topometric Map Construction

Fig. 6 shows an example of the built topometric map. Layout region vertices, object vertices with bounding boxes, and entrance vertices connecting regions are shown with edges representing relationships. A planned navigable path is shown in the graph from an observation in family room to the TV room sofa. We also compare the graph structure with graph-based method shown in Table IV, indicating that Topo-Field employs both object and region vertices with high accuracy.

### D. Ablation Study

Fig. 7 and Table V show the ablation of our neural field LOP encoding strategy and feature fusion where: 1) CLIP-Field [9] means the origin implementation of CLIP-Field. 2) Based on CLIP-Field, Baseline1 integrates region ground truth labels and encodes them with CLIP and S-BERT to generate target embeddings. These embeddings are employed as additional supervision of object pixel features besides the origin object embeddings to train $F$. 3) Instead of using vision-language encoder to encode the region text, Baseline2 encodes the origin image as the vision-language embedding for context, which takes the background pixels into account with the region labels. 4) Topo-Field, as current implementation, further considers the context of the layout when supervising the object label semantics, formatted as "object in region". These four main versions of our numerous iterations of trying are listed as examples to show our work on the neural field encoding of LOP association.

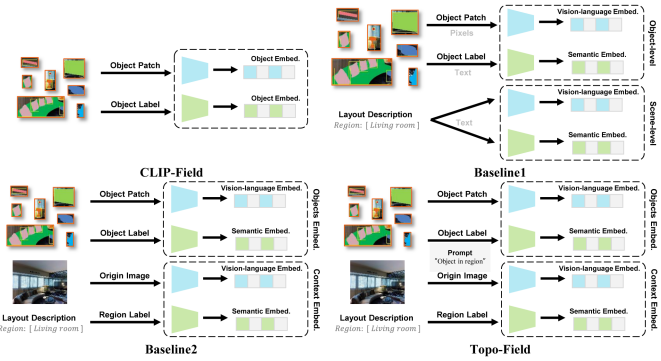


Fig. 7. Ablation of our LOP information encoding and feature fusion strategy for target features. Compared with CLIP-Field, Baseline1 learns region embeddings with additional supervision. Baseline2 encodes image into vision-language embeddings instead of singular text embeddings. Topo-Field further refines the object semantic label.

TABLE V
ABLATION OF TARGET FEATURE PROCESSING PIPELINE OF THE NEURAL FIELD CONSTRUCTION

| Methods | Scene1 | Scene2 | Scene3 | Scene4 |
|---|---|---|---|---|
| CLIP-Field | 0.242 | 0.165 | 0.130 | 0.142 |
| Baseline1 | 0.865 | 0.887 | 0.872 | 0.879 |
| Baseline2 | 0.872 | 0.891 | 0.875 | 0.886 |
| Topo-Field(Ours) | **0.886** | **0.900** | **0.884** | **0.894** |

The average region prediction accuracy of sampled points from different scenes on the Matterport3D dataset is used as the metric.

## VI. CONCLUSION AND LIMITATIONS

We propose a brain-inspired Topo-Field, which integrates Layout-Object-Position (LOP) associations into a neural field and constructs a topometric map from the learned field for hierarchical robotic scene understanding. However, there are some limitations: 1) Querying and path planning are currently implemented using traditional methods (e.g. A*). Future work

will explore more advanced path planning. 2) Current topological map is built on the static environment assumption. Future research will focus on updating and editing the topometric map to accommodate environmental changes. 3) Current region information needed for neural field training relies on human-labor, although the annotation process is easy. The region could be automatically annotated by reasoning the room contents in the future work.

## REFERENCES


[1] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[2] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart, "Topomap: Topological mapping and navigation based on visual SLAM maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3818–3825.

[3] S. Ullman, "The interpretation of structure from motion," *Proc. Roy. Soc. London. Ser. B. Biol. Sci.*, vol. 203, no. 1153, pp. 405–426, 1979.

[4] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 15–22.

[5] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 76a.

[6] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[7] S. Zhi, T. Laidlow, S. Leutenegger, and A. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 15838–15847.

[8] Z. Fan, P. Wang, Y. Jiang, X. Gong, D. Xu, and Z. Wang, "NeRF-SOS: Any-view self-supervised object segmentation on complex scenes," in *Proc. 11th Int. Conf. Learn. Representations*, 2023.

[9] N. Muhammad, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "CLIP-fields: Weakly supervised semantic fields for robotic memory," in *Proc. Robot.: Sci. Syst.*, 2023.

[10] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10608–10615.

[11] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language embedded radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19729–19739.

[12] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," in *Proc. Int. Symp. Exp. Robot.*, Chiang Mai, Thailand, 2023, pp. 105–117.

[13] K. Jatavallabhula et al., "ConceptFusion: Open-set multimodal 3D mapping," in *Proc. Conf. Robot., Sci. Syst*, 2023.

[14] S. Peng, K. Genova, C. M. Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, "OpenScene: 3D scene understanding with open vocabularies," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 815–824.

[15] C. Gomez et al., "Hybrid topological and 3D dense mapping through autonomous exploration for large indoor environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9673–9679.

[16] E. C. Tolman, "Cognitive maps in rats and men," *Psychol. Rev.*, vol. 55, no. 4, 1948, Art. no. 189.

[17] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat," *Brain Res.*, vol. 34, pp. 171–175, 1971.

[18] E. T. Rolls, A. Treves, R. G. Robertson, P. Georges-François, and S. Panzeri, "Information about spatial view in an ensemble of primate hippocampal cells," *J. Neuriophysiol.*, vol. 79, no. 4, pp. 1797–1813, 1998.

[19] P. A. LaChance, T. P. Todd, and J. S. Taube, "A sense of space in postrhinal cortex," *Science*, vol. 365, no. 6449, 2019, Art. no. eaax4192.

[20] T. Zeng, B. Si, and J. Feng, "A theory of geometry representations for spatial navigation," *Prog. Neurobiol.*, vol. 211, 2022, Art. no. 102228.

[21] J. Yang, R. Ding, Z. Wang, and X. Qi, "RegionPLC: Regional point-language contrastive learning for open-world 3D scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 19823–19832.

[22] D. Maggio et al., "Clio: Real-time task-driven open-set 3D scene graphs," *IEEE Robot. Automat. Lett.*, vol. 9, no. 10, pp. 8921–8928, Oct. 2024.

[23] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3D scene graphs for language-grounded robot navigation," in *Proc. Conf. Robot., Sci. Syst.*, 2024.

[24] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[25] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, Hong Kong, Nov. 2019, pp. 3982–3992.

[26] Q. Zhang, *Autonomous Indoor Exploration and Mapping Using Hybrid Metric/Topological Maps*. Montreal, QC, Canada: McGill Univ., 2015.

[27] Q. Zhang, I. Rekleitis, and G. Dudek, "Uncertainty reduction via heuristic search planning on hybrid metric/topological map," in *Proc. 12th Conf. Comput. Robot Vis.*, 2015, pp. 222–229.

[28] Q. Gu et al., "ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 5021–5028.

[29] S. -C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7515–7525.

[30] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Proc. Robot.: Sci. Syst.*, 2022.

[31] Z. He, H. Sun, J. Hou, Y. Ha, and S. Schwertfeger, "Hierarchical topometric representation of 3D robotic maps," *Auton. Robots*, vol. 45, no. 5, pp. 755–771, 2021.

[32] J. L. Schönberger and J. -M. Frahm, "Structure-from-motion revisited," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[33] A. Chang et al., "Matterport3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 667–676.

[34] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, 2022, Art. no. 102.

[35] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 350–368.

[36] Z. Zhu et al., "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12786–12796.

[37] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *Proc. Int. Conf. Learn. Representations*, 2022.