Learning Generalizable Feature Fields for Mobile Manipulation

Ri-Zhao Qiu*[1], Yafei Hu*[1,2], Ge Yang[3,4], Yuchen Song[1], Yang Fu[1], Jianglong Ye[1], Jiteng Mu[1], Ruihan Yang[1],
Nikolay Atanasov[1], Sebastian Scherer[2], Xiaolong Wang[1]
*equal contribution
[1]UC San Diego, [2]CMU, [3]MIT, [4]IAIFI
https://geff-b1.github.io

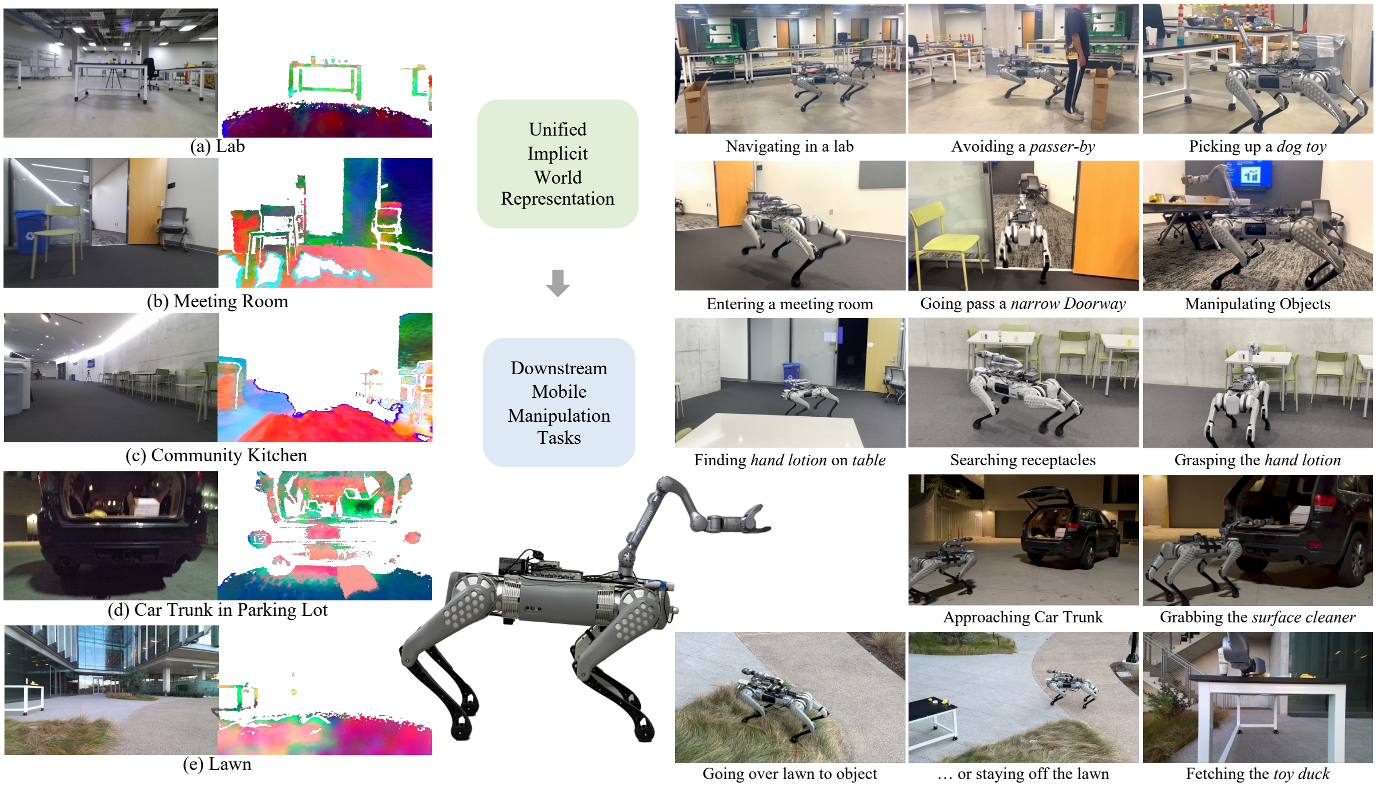
(a) Lab
(b) Meeting Room
(c) Community Kitchen
(d) Car Trunk in Parking Lot
(e) Lawn
Unified Implicit World Representation
Downstream Mobile Manipulation Tasks
Navigating in a lab
Avoiding a *passer-by*
Picking up a *dog toy*
Entering a meeting room
Going pass a *narrow Doorway*
Manipulating Objects
Finding *hand lotion* on *table*
Searching receptacles
Grasping the *hand lotion*
Approaching Car Trunk
Grabbing the *surface cleaner*
Going over lawn to object
… or staying off the lawn
Fetching the *toy duck*

Fig. 1: **GeFF**, **Ge**neralizable **F**eature **F**ields, provide unified implicit scene representations for both robot navigation and manipulation in real-time. We demonstrate the efficacy of GeFF on **open-world mobile manipulation** and **semantic-aware navigation** under diverse real-world scenes ((a) working in a lab where a person walks in, (b) entering a meeting room with narrow entrance, (c) operating in a community kitchen with various furniture, (d) grasping objects from a car trunk in a parking lot, and (e) semantic-aware navigation near a lawn outside of a building). The visualization of the feature fields is obtained by PCA of neural-rendered features. For best illustration, we animate feature fields built in real-time on the website.

***Abstract*—An open problem in mobile manipulation is *how to represent objects and scenes* in a unified manner so that robots can use it both for navigating in the environment and manipulating objects. The latter requires capturing intricate geometry while understanding fine-grained semantics, whereas the former involves capturing the complexity inherited to an expansive physical scale. In this work, we present GeFF (Generalizable Feature Fields), a scene-level generalizable neural feature field that acts as a *unified* representation for both navigation and manipulation that performs in real-time. To do so, we treat generative novel view synthesis as a pre-training task, and then align the resulting rich scene priors with natural language via CLIP feature distillation. We demonstrate the effectiveness of this approach by deploying GeFF on a quadrupedal robot equipped with a manipulator. We evaluate GeFF's ability to generalize to open-set objects as well as running time when performing open-vocabulary mobile manipulation in dynamic scenes.**

I. Introduction

Building a personal robot that can assist with common chores has been a long-standing goal of robotics [1, 2, 3]. This paper studies the task of open-vocabulary navigation and picking, where a robot needs to navigate through diverse scenes to pick up objects based on language instructions. This task, while seemingly easy for humans, remains quite challenging for autonomous robots. We humans achieve such tasks by understanding the layout of rooms and the affordances of objects without explicitly memorizing every aspect. However, when it comes to autonomous robots, there does not exist a unified scene representation that captures geometry and semantics for both navigation and manipulation tasks.

Recent approaches in navigation seek representations such as geometric maps (with semantic labels) [4, 5] and topological maps [6, 7] to handle large-scale scenes, but are not well integrated with manipulation requirements. Manipulation, on the other hand, often relies on continuous scene representation such as implicit surfaces or meshes [8] to compute precise grasping poses, which are not typically encoded in navigation representations. More importantly, interpreting semantic task instructions requires grounding of concepts with respect to geometric and semantic ***features*** of the environment. Such discrepancy in representations leads to unsatisfactory performance [9] in complex tasks that involve multiple visuomotor skills. Performing coherent open-vocabulary perception for both navigation and manipulation remains a significant challenge.

To this end, we present a novel ***scene-level*** **Ge**neralizable **F**eature **F**ield (**GeFF**) as a ***unified*** representation for navigation and manipulation. This representation is trained with neural rendering using Neural Radiance Fields (NeRFs) [10]. Instead of fitting a single NeRF with a static scene, our representation can be updated in real-time as the robot moves and the surroundings change. Inspired by recent works in **Gen**eralizable NeRFs (Gen-NeRFs) [11, 12], we train our representation with an encoder, which allows one feed-forward pass in the network to update the scene representation during inference. Besides being a unified representation, GeFF stands out with two more advantages: (i) GeFF is able to decode multiple 3D scene representations from a posed RGB-D stream, including SDF, mesh, and pointcloud, and (ii) by performing feature distillation from a pre-trained Vision-Language Model (VLM), e.g., CLIP [13], the representation not only contains geometric information but also language-conditioned semantics. These three key factors mitigate the discrepancy as discussed in the previous paragraph.

We demonstrate GeFF using a quadrupedal mobile manipulator to execute object discovery and manipulation tasks specified using language instructions. Our mobile manipulation system works as follows. First, the robot scans part of the scene which includes the target objects using an RGB-D camera and constructs a 3D representation using GeFF. At the same time, GeFF enables constructing a 3D feature field via feature distillation. The robot can then identify the goal object by searching in the feature field given language instructions. With the 3D map and an object goal, the robot can perform semantic-aware planning for navigation to reach and grasp target objects. As the robot moves in the scene, the RGB-D streams are fed into GeFF to extract 3D semantic features, and the pre-computed feature field is updated in real-time. This brings two benefits: (i) when the object arrangement (goal object or surroundings) changes in the scene, we can update the map in real-time and perform re-planning; (ii) as the robot moves closer to the object, GeFF can provide a more detailed description of the object given higher resolution inputs, which is essential for grasping.

We experiment with a Unitree B1 quadrupedal robot where a Z1 robot arm is attached on top, as shown in Fig. 1. We perform mobile manipulation with this robot in diverse environments where the robot needs to navigate to different receptacles, avoid dynamic obstacles, plan semantically to stay on walkways away from the lawn, and search and pick up objects in an open-vocabulary manner. We show that using the GeFF representation significantly improves over baselines using standard NeRF with feature distillation (e.g., LeRF [14]): GeFF achieves an average 52.9% success rate while LeRF achieves an average 30.7% success rate in mobile manipulation. We further perform ablations in simulation to validate the effectiveness of our approach. We plan to release the pre-trained models and the source code.

II. Related Work

Generalizable Neural Radiance Fields. Generalizable Neural Radiance Fields extend conventional Neural Radiance Fields' ability to render highly-detailed novel views to scenes that come with just one or two images [11, 15, 16, 17, 18, 19, 20, 12]. They replace the time-consuming optimization of weights for each scene with a single feed-forward process through a network. Existing works [19, 21, 22] mainly focus on synthesizing novel views. Our focus is to use novel view synthesis via generalizable neural fields as a generative pre-training task. At test time, we use the produced network for real-time semantic and geometric inference in a robotic mobile manipulation setting.

Feature Distillation in NeRF Beyond just synthesizing novel views, recent works [14, 23, 24, 12] have also attempted to combine NeRF with 2D features via feature distillation from 2D vision foundation models [13, 25, 26, 27] to 3D space to empower neural fields with semantic understanding of objects [23, 24, 12], scenes [14, 28] and downstream robotic applications [29, 28]. Nonetheless, these works cannot be easily adapted for mobile manipulation due to the expensive per-scene optimization scheme [14, 23] or restrictions to object-level representations [12]. Most closely related to our work, LERF [14] and F3RM [28] distill CLIP features to create scene representations that can be queried with natural language. F3RM adapts the feature fields for tabletop manipulation. Nonetheless, both LERF and F3RM require expensive per-scene optimization, which takes up to 40 minutes [14] to create a scene-level representation. Such an inefficiency hinders practical downstream applications on mobile robots. In stark contrast, our GeFF runs in real-time on mobile robots.

Object Navigation and Mobile Manipulation. Object navigation involves controlling a robot to navigate in the environment and to find target objects. Existing object navigation methods tackle this problem via modular approaches,

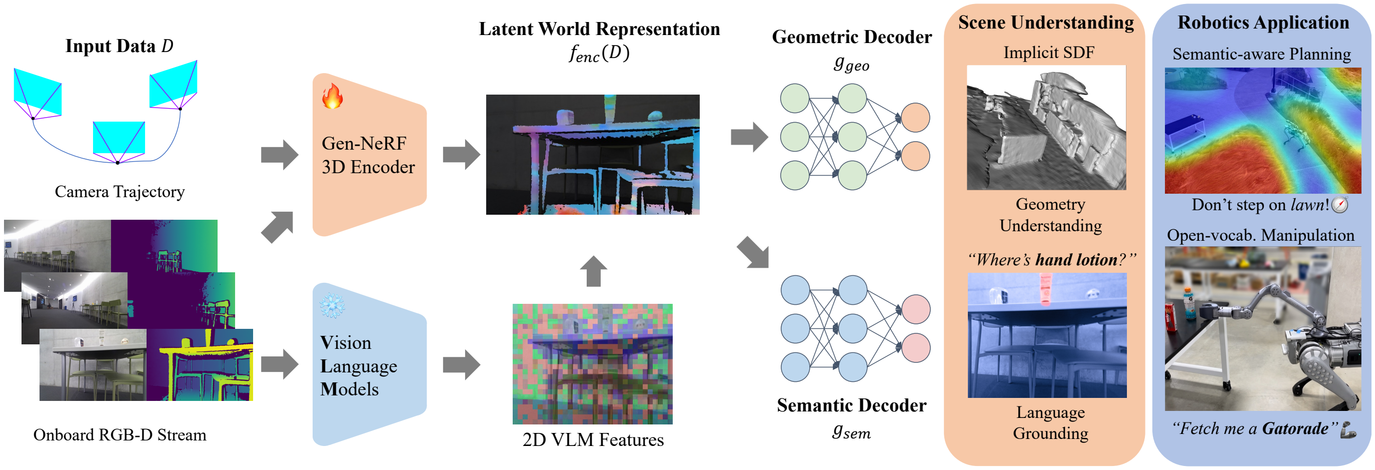

Fig. 2: Pre-trained as a generalizable NeRF encoder, **GeFF** provides unified scene representations for robot tasks from onboard RGB-D stream, offering both real-time geometric information for planning and language-grounded semantic query capability. Compared to LERF [14], GeFF runs in real-time without costly per-scene optimization, which enables many potential robotics applications. We demonstrate the efficacy of GeFF on **open-world language-conditioned mobile manipulation**. Feature visualizations are done by running PCA on high-dimensional feature vectors and normalizing the 3 main components as RGB.

using vision and language foundation models, scene graphs, etc. [30, 31, 32, 33, 34, 35, 36, 37], or via learning-based methods [38, 39, 40, 41, 42]. However, most end-to-end learning-based methods were only tested in constrained simulation environments. In addition, recently Gervet et al. [31] found that these end-to-end methods show poor generalization in real-world environments. Inspired by this finding, we follow the modular approach and combine GeFF with a classical motion planner and controller for real-world mobile manipulator deployment.

Beyond navigating to object goals, mobile manipulation requires a synergistic combination of navigation and manipulation [43, 44, 45, 9, 46, 47]. Previous works include learning-based methods [44, 48, 49, 45, 50, 51], and classical approaches based on motion planning [52, 53] or visual servoing [54]. Nonetheless, these works are constrained to a closed-set setting, meaning that they only work with a pre-defined range of objects that have seen during training. In contrast, our work operates on an open set of objects in both manipulation (specifying objects to manipulate) and navigation (instructing robots to avoid objects). Most recent works like HomeRobot [9] show open-vocabulary capabilities but have demonstrated only relative low success rate in small-scale real-world environments. In comparison with our approach, existing techniques lack a rich and unified 3D semantic and geometric representation of the environment to support integrated robot navigation and manipulation.

Open-Vocabulary Scene Representations. There have been some recent works [55, 56, 33, 57, 58] that leverage 2D foundation vision models to build open-vocabulary 3D representations. These methods project predictions from large-scale models such as CLIP [13] or SAM [59] directly onto explicit representations (point-based or voxel-based). As the number of features stored at each location increases, these explicit representation-based methods become harder to scale and are mostly limited to room-scale environment. GeFF, on the other hand, builds a *latent and unified representation* that conceptually scale to larger environments. A concurrent work, OK-Robot [56], uses voxel-based representations to perform open-vocabulary mobile manipulation, which is most related to our work among existing methods. In turn, GeFF demonstrates ability to operate in both room-scale environment and larger-scale outdoor environment with the its perceptive capability and traversability of quadruped robots.

III. Problem Formulation and Background

A. Problem Statement

Let Ω be the space of RGB-D images. Consider N posed RGB-D frames $\mathcal{D} = \{(F_i, \mathbf{T}_i)\}_{i=1}^N$ obtained from a mobile robot equipped with an RGB-D camera, where $F_i \in \Omega$ is the i-th RGB-D frame and $\mathbf{T}_i \in \mathbf{SE}(3)$ is the camera extrinsics. Our goal is to create a ***unified*** scene representation that captures geometric and semantic properties for robot navigation and manipulation from $\mathcal{D}$. More specifically, we aim to design an encoding function $f_{enc}(\cdot) : (\Omega \times \mathbf{SE}(3))^N \mapsto \mathbb{R}^{N \times C}$ that compresses $\mathcal{D}$ to a latent representation, and decoding functions $g_{geo}(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^{N \times C} \mapsto \mathbb{R}^m$ and $g_{sem}(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^{N \times C} \mapsto \mathbb{R}^n$ that decode the latents into different geometric and semantic features at different positions in 3D space. Here, C, m, and n are the dimensions of the latent representation, geometric feature, and semantic feature, respectively. These geometric and semantic features can then serve as the input to the downstream planner. We aim to design these three functions to meet the following criteria.

- **Unified.** The encoded scene representation $f_{enc}(\mathcal{D})$ is ***sufficient*** for both geometric and semantic query (*i.e.*, g_{geo} and g_{sem} are conditioned on $\mathcal{D}$ only via $f_{enc}(\mathcal{D})$).
- **Incremental.** The scene representation supports efficient incremental addition of new observations, (*i.e.*, $f_{enc}(\mathcal{D}_1 \cup \mathcal{D}_2) = f_{enc}(\mathcal{D}_1) \oplus f_{enc}(\mathcal{D}_2)$)

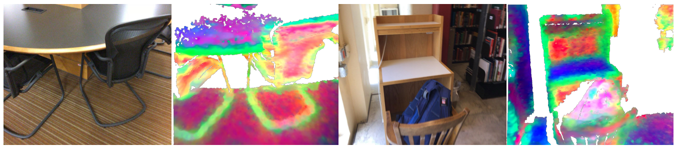

Fig. 3: **Generalizable Feature Fields Acquire Geometric and Semantic Priors.** RGB images are input views from ScanNet [60]. Color images are PCA visualizations of feature volume projected to the input camera view encoded by an RGB-D Gen-NeRF [61] encoder. Note how semantically similar structures acquire similar features.

- **Continuous.** To support hierarchical, coarse-to-fine-grained robotic motion planning and control, the query function should be continuous and queryable at any coordinate (*e.g.*, $g_{geo}(\mathbf{x}, f_{enc}(\mathcal{D}))$, where $\mathbf{x} = (x, y, z)$ is a location in 3-D space).
- **Implicit.** The encoded latents $f_{enc}(D)$ should be organized in a sparse implicit representation to enable more efficient scaling to large scenes than storing $\mathcal{D}$.
- **Open-world.** The semantic knowledge from g_{sem} should be open-set and aligned with ***language***, so the robot can perform open-world perception. That is, the feature vector output from g_{sem} lies in a multi-modality that aligns text and vision (*e.g.,* CLIP [13]).

In this paper, we build GeFF upon generalizable NeRFs to satisfy all of these requirements.

B. Background: NeRF and Generalizable NeRF

Given a coordinate $\mathbf{x} \in \mathbb{R}^3$ and a unit-vector viewing direction $\mathbf{d} \in \mathbf{S}^2$, the original NeRF [10] adopts two parameterized networks, a density mapping $\sigma_\theta(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}[0, 1]$, which predicts the probability that $\mathbf{x}$ is occupied, and a color mapping $\mathbf{c}_\omega(\mathbf{x}, \mathbf{d}) : \mathbb{R}^3 \times \mathbf{S}^2 \to \mathbb{R}^3$ which maps coordinate and viewing direction to color.

Consider a ray $\mathbf{r}$ from a camera viewport with camera origin $\mathbf{o}$ and direction $\mathbf{d}$, which is conditioned by the camera extrinsics $\mathbf{T}$ and intrinsics $\mathbf{K}$. NeRF estimates color along $\mathbf{r}$ by

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\alpha_\theta(\mathbf{r}(t))\mathbf{c}_\omega(\mathbf{r}(t), \mathbf{d})\mathrm{d}t\,, \tag{1}$$

where t_n and t_f are minimum and maximum distances bounding depth, $T(t) = \exp(-\int_{t_n}^{t} \sigma_\theta(s)\mathrm{d}s)$ is the transmittance which denotes the observed occupancy so far to avoid rendering voxels behind objects, and $\alpha_\theta(r(t))$ being the opacity value at $r(t)$ (original NeRF [10] uses $\alpha_\theta = \sigma_\theta$).

NeRF then optimizes θ and ω w.r.t. color by

$$\mathcal{L}_{col}(\theta, \omega) = \sum_{\mathbf{r} \in \mathcal{R}} \left|\left|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\right|\right|_2^2\,, \tag{2}$$

where $\mathbf{C}(\mathbf{r})$ denotes the ground truth color of the ray $\mathbf{r}$, and $\mathcal{R}$ is a set of randomly sampled ray for training. Note that the training process ***starts from scratch*** for every scene and may take hours. To avoid costly per-scene optimization, generalizable NeRFs [11, 12] propose to ***condition*** the novel view on input frames instead of optimizing the underlying parameters. During its training time, generalizable NeRFs learn to ***incorporate scene priors*** into its encoder. More concretely, the occupancy and the radiance networks are given by

$$\sigma_\theta(\mathbf{x}, \mathcal{D}) = g_\sigma(\mathbf{x}, f_{enc}(\mathcal{D})) \tag{3}$$

$$\mathbf{c}_\omega(\mathbf{x}, \mathbf{d}, \mathcal{D}) = g_{\mathbf{c}}(\mathbf{x}, \mathbf{d}, f_{enc}(\mathcal{D}))\,, \tag{4}$$

where g_σ and $g_{\mathbf{c}}$ are MLPs that predict density and color, f_{enc} being a neural encoder. Note that parameters are learned during pre-training. During the testing stage, novel views are efficiently rendered in a single-pass manner.

IV. GeFF for Mobile Manipulation

We describe our approach, Generalizable Feature Fields (GeFF), and ways we apply it to mobile manipulation tasks in the following subsections. An overview of our method is shown in Fig. 2.

A. Learning Scene Priors via Neural Synthesis

Generalizable neural radiance fields acquire rich geometric and semantic priors by learning to synthesize novel views in many scenes [61, 62, 12]. To illustrate this point, we offer a motivating example in Fig. 3, where we render the intermediate feature volume from an RGB-D Gen-NeRF encoder [61] trained to synthesize novel views on the ScanNet [60] dataset. The colors correspond to the principal components of the latent features. We observe separations between objects and the background, despite no explicit semantic supervision were provided during training (only the RGB-D views and the camera poses were available). This example highlights the potential of using neural synthesis as a generative pre-training task for learning scene priors.

We propose two types of training signals using both the 2D foundation models and depth to provide supervision.

Supervision (i): Language-Alignment via Feature Distillation. Although we have shown that Gen-NeRF encoders implicitly encode geometric and semantic cues, the representation is less useful if it is not ***aligned*** to other feature modalities, such as language. In order to further enhance the representation capability of GeFF, we propose to use knowledge distillation to transfer learned priors from 2D vision foundation models and align the 3D representations with them. To the best of our knowledge, GeFF is the ***first*** approach that combines scene-level generalizable NeRF with feature distillation. In stark contrast to previous attempts [23, 14, 12], GeFF both works in relatively large-scale environments and runs in real-time, making it a powerful perception method for mobile manipulation.

Specifically, we build a feature decoder $g_{sem}(\mathbf{x}, f_{enc}(D))$ on top of the latent representation, which maps a 3D coordinate to a feature vector. The output of g_{sem} is trained to be aligned with the embedding space of a teacher 2D vision foundation model, termed $f_{teacher}$. Note that g_{sem} is non-directional. Intuitively, the semantic properties of an object should be view-independent (*e.g.,* a cup is a cup regardless of the viewing directions). Similar to color rendering in Eq. 1, we can render 2D features for pre-training via

$$\hat{\mathbf{F}}(r) = \int_{t_n}^{t_f} T(t)\alpha(r(t))g_{sem}(\mathbf{r}(t), f_{enc}(\mathcal{D}))\mathrm{d}t\,. \tag{5}$$

To further enhance the fidelity of the 3D scene representation, we use the 2D features of the input views computed by the teacher model as an auxiliary input to f_{enc}, which is

$$f_{enc}(D) = \textsc{ConCat}\left(\hat{f}_{enc}(\mathcal{D}), f_{teacher}(\mathcal{D})\right)\,, \tag{6}$$

where $\hat{f}_{enc}$ is a trainable encoder and $f_{teacher}$ is a pre-trained vision model with frozen weights. We ablate the effect of the auxiliary input in the experiments. The final feature rendering loss is then given by

$$\mathcal{L}_{feat} = \sum_{\mathbf{r}\in\mathcal{R}} \left|\left|\mathbf{F}(\mathbf{r}) - \hat{\mathbf{F}}(\mathbf{r})\right|\right|_2^2 \tag{7}$$

where $\mathbf{F}$ is the reference ground-truth features obtained by running foundation models on ground-truth novel views. Compared to previous works that use view-dependent features [14, 28], one strength of our training scheme is that the encoded features are view-independent, which makes it a favorable representation for downstream motion planners that often require 3D information.

Model for Distillation. Our proposed feature distillation method for scene-level generalizable NeRFs is generic and can be applied to many 2D vision foundation models such as Stable Diffusion [27], SAM [59], and DINO [26]. In this work, however, since we are interested in tasking robots to perform open-vocabulary mobile manipulation, we need to choose a vision foundation model that is aligned with language. Therefore, we choose MaskCLIP [63] as $f_{teacher}$, which is a variant of CLIP [13] that exploits a reparametrization trick [63, 28] to transform the output of CLIP from a single feature vector to a feature map aligned with the CLIP text encoders. Though the 2D feature maps from MaskCLIP are coarse (illustrated in Fig. 2), it is efficient enough to run at real time on mobile robots and we show qualitatively that GeFF learns to reconstruct fine details from multiple views.

Handling Language Query. Following standard protocols [14, 28], GeFF takes in positive text queries along with a few negative text queries (*e.g., wall* and *ceiling*). These text queries are encoded using CLIP's text encoders, which send texts to an embedding space that is aligned with the distill features. To rate the similarity of a coordinate with a positive text query, we use cosine similarity to compute the rendered feature with each text query. A temperatured softmax is then applied to the cosine similarity to form a probability distribution. Finally, we sum up the probabilities of positive queries to formulate the similarity score.

Supervision (ii): Depth Supervision via Neural SDF. Inspired by [18, 64, 61], we introduce a signed distance network $s(\mathbf{x}) = g_{geo}(\mathbf{x}, f_{enc}(\mathcal{D}))$ to encode depth information. Doing so has two advantages over previous work [11]: 1) it leverages depth information to ***efficiently*** resolve scale ambiguity for building scene-level representation, rather than restricted to object-level representation, and 2) it creates a continuous implicit SDF surface representation, which is a widely used representation for robotics applications such as computing collision cost in motion planning [64].

To provide supervision for g_{geo} during pre-training, we follow iSDF [64] and introduce an SDF loss $\mathcal{L}_{\text{sdf}}$ and an Eikonal regularization loss [65] $\mathcal{L}_{\text{eik}}$ to ensure smooth SDF values. The main difference with iSDF [64] is that we condition g_{geo} with $f_{enc}(\mathcal{D})$, which ***does not require optimization for novel scenes***. Instead of using a density network, we represent the opacity function α in Eq. 1 using $s(\mathbf{x})$

$$\alpha(r(t)) = \textsc{Max}\left(\frac{\sigma_s(s(\mathbf{x})) - \sigma_s(s(\mathbf{x}+\Delta))}{\sigma_s(s(\mathbf{x}))}, 0\right)\,, \tag{8}$$

where σ_s is a sigmoid function modulated by a learnable parameter s. The depth along a ray $\mathbf{r}$ is then rendered by

$$\hat{\mathbf{D}}(r) = \int_{t_n}^{t_f} T(t)\alpha(r(t))d_i\mathrm{d}t\,, \tag{9}$$

where d_i is the distance from current ray marching position to the camera origin. Similar to Eq. 2, the rendered depth can be supervised via

$$\mathcal{L}_{depth} = \sum_{\mathbf{r}\in\mathrm{R}} \left|\left|\mathbf{D}(\mathbf{r}) - \hat{\mathbf{D}}(\mathbf{r})\right|\right|_2^2\,. \tag{10}$$

Implementation Details. For a single posed RGB-D frame, f_{enc} follows existing works in 3D encoder [66, 67] and encodes a single view to a 3D volume of shape $\mathbb{R}^{M\times C}$. Here, $M = 512$ is a set of sparse points obtained via the farthest point sampling and C is the feature dimension of each point. The obtained points are also transformed to the world frame using camera intrinsics and extrinsics to build a consistent world representation from multi-frame observations.

Features of these M points are obtained by using PointCov [68] on the downsampled 3D points and interpolating a dense feature map from a ResNet-50 [69] encoder. For a specific point query in generalizable NeRF decoding, f_{enc} interpolates features from nearby K points. The decoders g_{sem} and g_{geo} are implemented as multi-layer MLPs. We will release the code for details and reproducibility.

Final Training Objective. Combining all the above equations, the total loss we used to train f_{enc} for a unified latent scene representation is given by

$$\begin{aligned}\mathcal{L} = \lambda_1\mathcal{L}_{col} + \lambda_2\mathcal{L}_{depth} + \lambda_3\mathcal{L}_{sdf} \\ + \lambda_4\mathcal{L}_{eik} + \lambda_5\mathcal{L}_{feat}\end{aligned} \tag{11}$$

where λ_i are hyperparameters used to balance loss scales.

B. Open-Vocabulary Mobile Manipulation

Scene Mapping with GeFF. As detailed in previous sections, GeFF encodes a single posed RGB-D frame to a latent 3D volume, which is represented as a sparse latent point cloud. Since per-frame 3D latent volume is back-projected to the world frame, we incrementally build the latent volume by concatenating per-frame observations. The camera pose used to

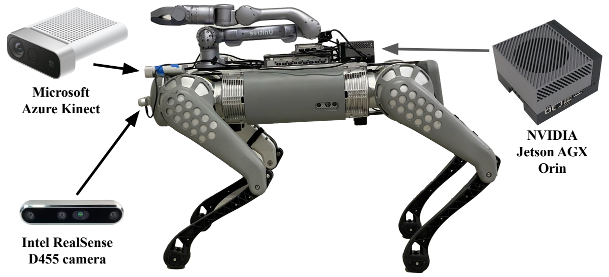

Fig. 4: The mobile manipulation platform. A 7-DOF Z1 robotic arm is mounted on top of the B1 quadrupedal robot. A forward facing Kinect sensor and RealSense camera are mounted at the front, and a Nvidia Orin onboard computer is mounted at the rear.

construct GeFF is provided by an off-the-shelf Visual-Inertial Odometry (VIO) method [70]. The latent 3D volume can then be decoded into geometric and semantic representations.

Decoded Representations. Though GeFF supports decoding to various representations, it is inefficient and impractical to generate all possible representations on-the-fly. For this work, we decode the latent representation into a point cloud and an occupancy map as geometric representations for navigation and manipulation. We then enhance basic units in these representations (*i.e.,* points and grid cells) with feature vectors from g_{sem}, which can be compared with language queries encoded by the CLIP [13] text encoder. The comparison results are per-point similarity scores with the higher-score responses being the points more similar to the description in the language instruction. For a visualization of the 3D map with score responses, please refer to Fig. 7.

GeFF for Navigation. We consider the navigation of the base quadrupedal robot as a 2D navigation problem due to the compute constraints of the robot. The 2D occupancy grid map which provides the traversability for navigation is downward projected by decoded 3D point cloud. The feature vector for each grid cell is created by averaging the feature vectors of related points. The location with the most points, whose feature vectors are top-k similar to the input language query, is chosen as the goal location. To support semantic-aware planning, we take in text queries of objects to avoid (*e.g., lawn*) and assign semantic affordances (*i.e.,* cost to traverse over a grid cell) to every grid cell using its similarity with the avoiding objects. The robot then uses a cost-aware A* planner to plan a set of waypoints to the goal location. We use a PD controller to track these waypoints.

Note that since GeFF runs in real-time, the goal location and obstacles are dynamically updated so that the robot can react to scene changes on the fly, which leads to the robot's ability to avoid previously unseen obstacles and to find new objects upon arriving at the specified receptacle. We evaluate GeFF's ability to handle scene change in V-D.

GeFF for Manipulation. After the robot arrives at the coarse goal location on the map, it aggregates the semantic point cloud decoded from GeFF with the same clustering algorithm to refine the centroid of the object. We then adjust the final pose of the robot base so that the centroid is within the configuration space of the end gripper of the robot manipulator. The robot then attempts to grasp the object via an open-push-close gripper action sequence with trajectories computed by a sample-based planner (OMPL planner [71]) to solve the Inverse Kinematic (IK) for the manipulator.

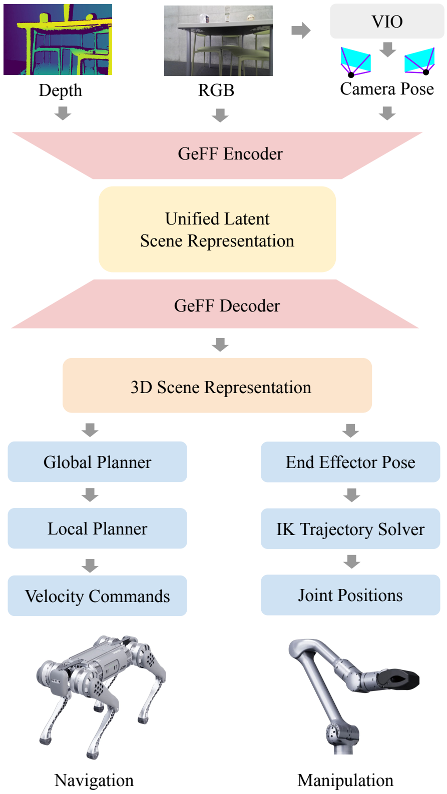

Fig. 5: Overview of our mobile manipulation autonomy system for navigation and manipulation.

V. Experiments

A. Experimental Setup

Real-world Evaluation. We deploy GeFF on a real robot to evaluate the efficacy of GeFF in performing various tasks such as open-vocabulary mobile manipulation and semantic-aware planning. In addition to working with the real robot, we also ablate design choices (*e.g.,* whether auxiliary inputs are used and teacher models) in simulation with Gazebo [72] and Habitat [73]. For quantitative experiments, we task the robot to perform mobile manipulation using language conditioning in 3 environments: a 25 m^2 lab with artificial obstacles, a 30 m^2 meeting room with chairs and a big rectangular table, and a 60 m^2 community kitchen with various furniture. We perform 3 trials on a total of 17 objects (6 miscellaneous objects for the lab, 5 office-related items for the meeting room, and 6 kitchen-related items for the kitchen) including 8 out-of-distribution categories that GeFF had ***not seen*** during pre-training on ScanNet [60]. For qualitative experiments, we test the robot's ability to navigate with language-conditioned semantic affordance, map the environment when a person walks into the scene, builds intricate geometry from multiple views, and entering narrow doorway.

		Lab Env.		Meeting Room Env.		Kitchen Env.		Overall	
Method	Latency	Nav. Succ.	Mani. Succ.	Nav. Succ.	Mani. Succ.	Nav. Succ.	Mani. Succ.	Nav. Succ.	Mani. Succ.
GeFF (Ours)	0.4s	**94.4%**	**61.1%**	**86.7%**	**53.3%**	**66.7%**	**44.4%**	**82.6%**	**52.9%**
GeFF no auxiliary	**0.2s**	55.6%	27.5%	60.0%	33.3%	38.9%	22.2%	51.5%	27.6%
LERF [14]	2 hrs	72.2%	44.4%	40.0%	20.0%	44.4%	27.8%	52.2%	30.7%

TABLE I: Mobile manipulation success rate categorized by navigation success rate (Nav. Succ.) and manipulation success rate (Mani. Succ.) under different environments with different methods. Our method consistently outperforms baseline methods.

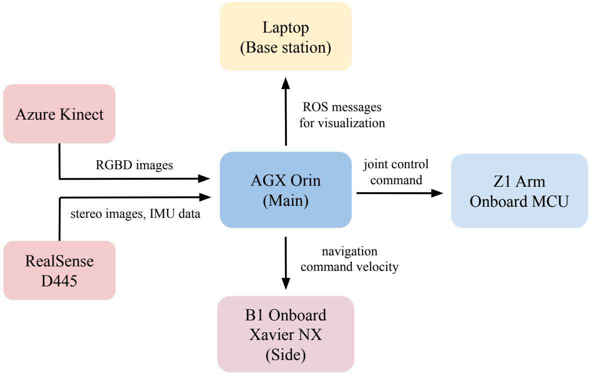

Fig. 6: Communication system setup based on ROS

Method	Scene Change	Lab	Meeting Rm.	Kitchen
GeFF (Ours)	✗	7/9	7/9	8/9
	✓	4/9	6/9	8/9
LERF [14]	✗	6/9	7/9	4/9
	✓	NA*	NA*	NA*

TABLE II: Mobile manipulation success rate under **scene change**, where objects are added to scenes after the robot builds an initial map. The results are reported on three objects (hand lotion, dog toy, and toy duck) over three trials per object. Note that LERF [14] requires costly per-scene optimization and thus can not handle scene change. Our method consistently outperforms the baselines.

Experimental Protocol. For each environment, we first manually drive the robot to explore the environment and build an initial representation of the environment. This process allows the robot to perceive the locations of fixed structures (*e.g.,* receptacles) and can be replaced by standard robotic exploration algorithms. During this process, the robot is ***agnostic*** to the final goal category. Instead, GeFF allows the robot to retrieve specific objects using language online. To make the task setting realistic and to demonstrate the capability of GeFF performing real-time mapping, we study a more challenging task setting where there may be ***scene changes*** between the initial mapping stage and the object retrieval stage.

Robotic Hardware Platforms. The hardware setup is shown in Fig. 4. We use Unitree B1 as the base robot and mount an Unitree Z1 arm on top of it. The robot is equipped with a Microsoft Kinect RGBD sensor to provide depth for GeFF, and an Intel RealSense D455 stereo camera to run VIO to localize the robot for GeFF. The onboard computer is an NVIDIA Jetson AGX Orin.

Metrics. Following protocols in existing work in mobile manipulation [49, 9, 44], we use success rate as the main evaluation metric. We define success for navigation as stopping the navigation process when the robot is within $1m$ of its front-facing goal object. We define success for mobile manipulation as navigating and grasping the specified object, lifting it off the surface, and holding it.

B. Autonomy System Overview

We use a modular approach and divide the autonomy system into the perception, motion planning, and control modules, which is illustrated in Fig. 5.

C. Communication System Setup

Based on ROS, we set up our communication system. We use the Nvidia Orin as the main computer, an (optional) Ubuntu 20.04 laptop as the base station to visualize the online results and performance, and the onboard Nvidia XavierNX of the robot as the side computer. We use the main computer to perform all the computations for the autonomy system, the base station to visualize the online results, and the side computer to receive velocity commands to execute actions for the robot base. The conversion from velocity commands to motor commands is done by Unitree SDK. The Z1 manipulator receives commands directly from the main computer. An overview of our communication system is shown in Fig. 6.

D. Quantitative Results on Real Robot

We evaluate GeFF's capability to help the robot perform open-set mobile manipulation. Since previous mobile manipulation methods either work on only a pre-defined set of objects [49] or have a low success rate and require specialized hardware [9], we use LERF [14], another feature field method, as the main baseline method. Since LERF is an RGB-only method with view-dependent features, we use metric poses estimated by the same VIO algorithm to alleviate scale ambiguity and select the point with maximum responses in features rendered from training views as the goal location. We evaluate the success rate of navigation and mobile manipulation.

Open-Set Mobile Manipulation. We test the mapping latency, navigation and manipulation success rates with a total of 17 objects in 3 different environments. The results are given in Tab. I. We compare a baseline method without using auxiliary input shown in Eq. 6, as well as LERF [14]. Most methods show the highest success rates on both tasks in the Lab, which is a controlled environment with consistent lighting. On the other hand, Kitchen, a realistic scene with

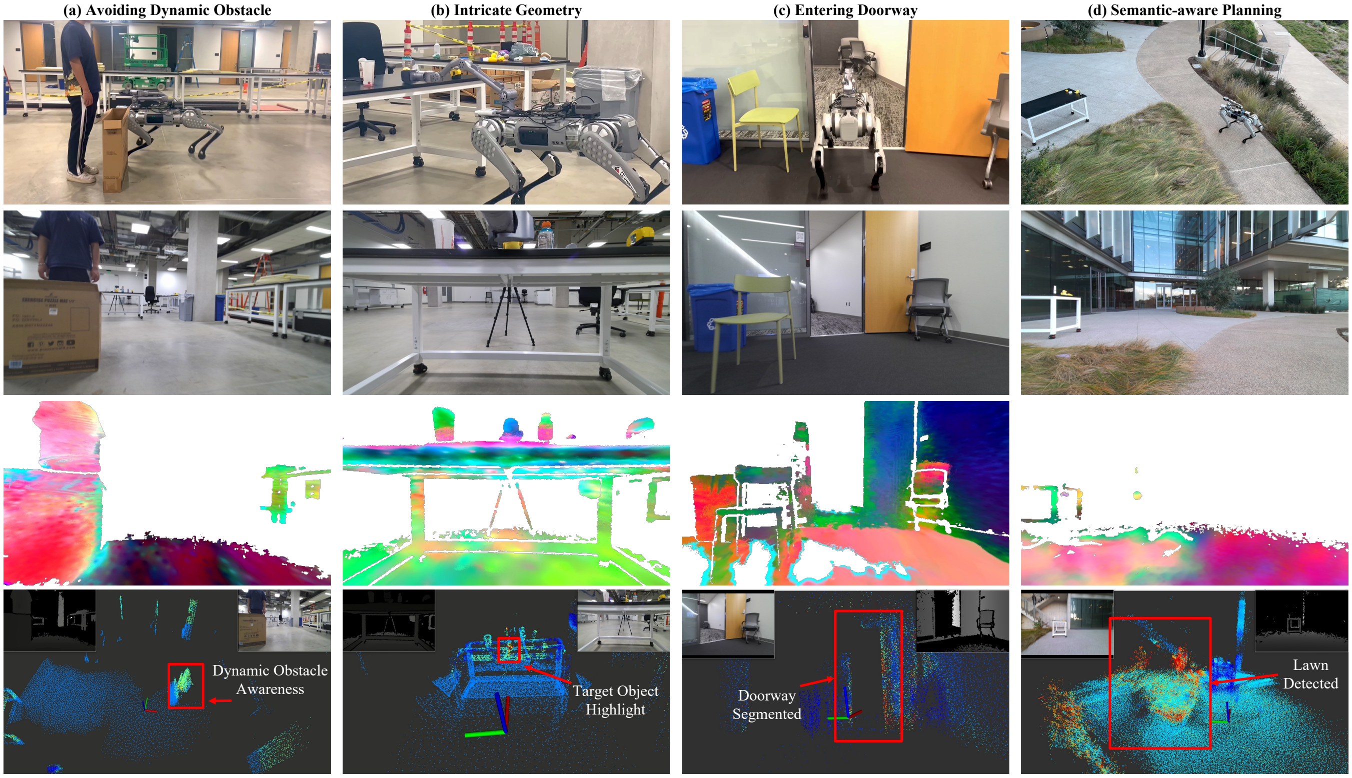

Fig. 7: Qualitative results of GeFF for diverse tasks. (a): using GeFF, the quadrupedal robot can build feature fields in real time, which allows detection of passer-by in real time. (b): building high-quality feature fields from multi-frame inputs. (c): entering a narrow doorway. (d): semantic-aware planning where the robot avoids lawn and stays on walkway. For best illustration, please refer to the website. (From top to bottom: side views of the robot; first-person views of the robot; PCA visualization of the first-person GeFF features; similarity response of feature point clouds to positive text queries).

challenging lighting conditions and complex scene layouts, poses challenges to all methods. Our method, GeFF augmented with auxiliary foundation model input, consistently achieves the best performance across all task settings.

Open-Set Mobile Manipulation with Scene Change. One notable characteristic of generalizable neural fields is that they do not require costly per-scene optimization. This is a desirable property for robotics applications, as we want robots to be able to respond to scene changes on the fly. In this part, we evaluate the performance of GeFF for open-set mobile manipulation where objects are added to the scenes after initial mapping. Specifically, we put objects on tables after the initial mapping stage. With new objects and language descriptions, the robot should be able to find the new objects, navigate to them, and then grasp them. We use a subset of objects from the main experiments and test the mobile manipulation success rates under three environments. Tab. II shows the results of open-set mobile manipulation with scene changes. LERF [14], being a conventional feature field method, requires costly per-scene optimization and is not applicable to online response to scene changes. On the other hand, our method, GeFF, successfully copes with scene changes.

E. Qualitative Results on Real Robot

In this section we provide qualitative results of GeFF to demonstrate different potential applications. In Fig. 7, we show qualitative results of dynamic obstacle avoidance ability in the lab environment, localizing objects with good geometric reconstruction by fusing multiple views, ability to go through a narrow doorway, and semantic-aware planning to avoid terrains that the robot should semantically avoid.

Dynamic Obstacle Avoidance. We first construct the map and while the robot moves toward the goal object, one person comes in front of the robot. From the first sub-figure in the 3rd row of Fig. 7 (a), we notice that GeFF recognizes people and can assign higher affordances in real time, which is challenging for per-scene optimization feature fields such as LERF [14].

Intricate Geometry. GeFF is capable of fusing features from multi-views to create a fine-grained semantic representation than a single-view projection. Illustrated in Fig. 7 (b) and Fig. 8, we demonstrate clear semantic and geometric boundaries from reconstructed objects.

Narrow Passage. Since GeFF can produce a ***fine-grained*** scene representation, it allows the robot to pass through a narrow doorway without collisions in the meeting room. This result is illustrated in Fig 7 (c).

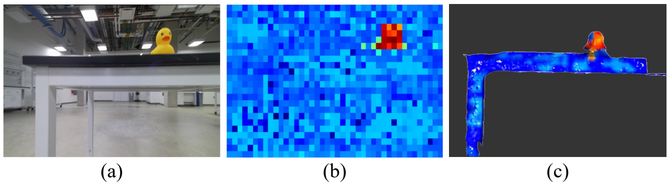

Fig. 8: GeFF fuses low-resolution coarse 2D features from multiple views for refinement. (a) A single RGB view of the object. (b) coarse 2D heatmap with text query 'toy duck' by CLIP [13]. (c) 3D heatmap from GeFF with clean and sharp object boundary. (Best viewed when zoomed in.)

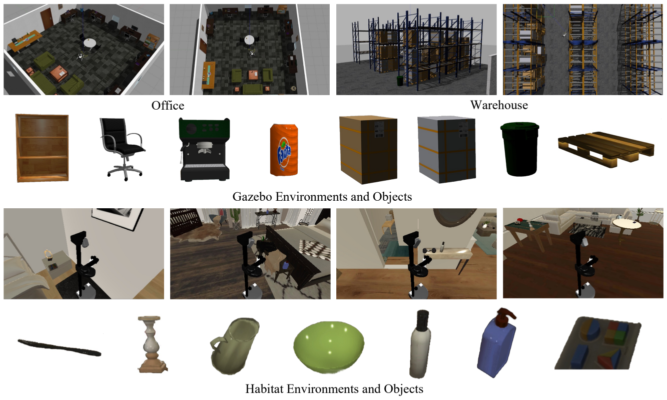
Office
Warehouse
Gazebo Environments and Objects
Habitat Environments and Objects

Fig. 9: Snapshots of the simulation environments and objects we use in our experiment. From this figure we are able to see the large domain gap between real-world and simulation, yet our method is able to handle the objects in simulation successfully.

Semantic-aware Planning. It is natural to integrate semantic affordances with the feature-enhanced point cloud and occupancy map provided by GeFF. In Fig. 7 (d), we test the capability of GeFF in an outdoor scene. With the traversability of the quadrupedal robot, the robot can directly step over lawn towards the goal object. However, it may be undesirable for robots to step on lawns. Therefore, we assign higher semantic affordances for grass surface, which encourages the robot to stay on the walkways.

F. Ablation Studies in Simulation

We use Gazebo [72] and Habitat [73] with OVMM extended data [73, 9] as our simulators. We use two environments (office and warehouse) with 8 objects in Gazebo, and use 7 scenes with 31 objects in Habitat. We also test the results with scene change in Gazebo. The scene change mainly consists of two categories: (1) adding dynamic obstacles and (2) add new target object. We use large objects such as bookshelf and big box to test the obstacle avoidance ability. We also test small-size objects like soda to indicate the open-set ability of GeFF. The results are shown in Tabs. III and IV.

We notice that the full GeFF with auxiliary inputs consistently outperforms the no auxiliary input version, which confirms the necessity of using foundation models during inference time. In addition, given perfect camera poses and depth, the success rate of goal localization and navigation are

		GeFF (Ours)		GeFF no aux.	
Env.	Scene Change	Goal Succ.	Nav. Succ.	Goal Succ.	Nav. Succ.
Office	✗	100.0%	75.0%	66.7%	25.0%
	✓	75.0%	66.7%	63.6%	8.3%
Warehouse	✗	88.9%	66.7%	66.7%	44.4%
	✓	88.9%	77.8%	66.7%	33.3%
Overall	✗	**94.4%**	**70.8%**	66.7%	34.7%
	✓	**81.3%**	**72.2%**	65.2%	20.8%

TABLE III: We compare with the baseline method without auxiliary (aux.) inputs of GeFF encoder. We evaluate the success rate of finding goal objects (Goal Succ.) as well as navigation success rate (Nav. Succ.). Note that in warehouse with scene change the navigation performs slightly better. This is due to the robot hit on the rack once while navigating and the case may happen when there are changes in scenes too.

Distill. Model	DinoV2 [26]	CLIP-Text [13]	CLIP-Image [13]
Nav. Succ.	67.6%	10.8%	27.0%

TABLE IV: Open-set object navigation results on habitat. We perform ablation study by distilling different vision foundation models.

good even when there is scene change, which identify the bottleneck for GeFF in real world as the localization accuracy and depth quality. In addition, we notice that the choice of vision foundation models matters. We use DinoV2 [26], which generates feature maps with higher resolution but is not aligned with languages, for goal navigation on Habitat. The feature of the goal object is specified using an image where DinoV2 [26] averages goal object features. For the Habitat OVMM dataset [9], DinoV2 shows good sucess rate, which hints future research to fuse features from multiple foundation models for GeFF. However, as illustrated in Fig. 9, the simulation environment has a large domain gap from real-world data. Therefore, in habitat, GeFF using CLIP and queried with text demonstrates unsatisfactory performance in finding goal objects.

G. Failure Analysis

We perform an in-depth analysis of failure cases to facilitate future research. Perception failure, including both the failure to precisely determine objects' goal locations or the inability to localize the robot, is the critical factor that leads to navigation failure in complex scenes (*i.e., kitchen*). In challenging lighting conditions, such as the kitchen where the spotlights may directly illuminate the localization camera, the VIO algorithm could be inaccurate due to its internal assumption of consistent lighting. This leads to imprecise world representation and undesired errors when solving inverse kinematics for manipulation. Future work could improve this by either 1) using an in-hand camera for manipulation or 2) designing a high-level policy that does not rely on accurate camera poses over the long horizon.

Besides perception failures, the manipulation also sometimes fails due to the current open-loop manipulation scheme, especially the end effector of the Z1 arm often fails on low-friction objects (*e.g.,* plastic drink bottles). Future work could include transforming the current manipulation solution to a close-loop system.

Failure Analysis in Scene Change. Scene changes pose unique challenges to perception. Though we can instruct the robot to navigate to furniture to map and manipulate new objects, the robot obtains only a *single view* of the object, which may lead to inaccurate representation unlike Fig. 8. Future work could design an exploration policy that attempts to automatically obtain multi-views of the object or learn priors for shape completion.

VI. Conclusion

In this paper, we present GeFF, a scene-level generalizable neural feature field with feature distillation from VLM that provides a unified representation for robot navigation and manipulation. Deployed on a quadrupedal robot with a manipulator, GeFF demonstrates zero-shot object retrieval ability in real-time in real-world environments. Using common motion planners and controllers powered by GeFF, we show competitive results in the open-set mobile manipulation tasks.

A promising future direction that may potentially address both issues is to learn a unified control policy on top of GeFF features to close the control loop, which can address the aforementioned failure reasons.

References

[1] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Advances in Neural Information Processing Systems*, 31:9094–9104, 2018.

[2] Joao MC Marques, Jing-Chen Peng, Patrick Naughton, Yifan Zhu, James S Nam, and Kris Hauser. Commodity telepresence with team avatrina's nursebot in the ana avatar xprize finals. In *ICRA 2023 2nd Workshop on Toward Robot Avatars*, 2023.

[3] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 2023.

[4] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P. How, and Luca Carlone. Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. *IEEE Transactions on Robotics (T-RO)*, 38(4):2022–2038, 2022.

[5] Arash Asgharivaskasi and Nikolay Atanasov. Semantic OcTree Mapping and Shannon Mutual Information Computation for Robot Exploration. *IEEE Transactions on Robotics (T-RO)*, 39(3):1910–1928, 2023.

[6] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *ICRA*, 2023.

[7] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. In *CORL*, 2023.

[8] Fan Wang and Kris Hauser. Stable bin packing of non-convex 3d objects with a robot manipulator. In *ICRA*, 2019.

[9] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023.

[10] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[11] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[12] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *ICCV*, 2023.

[13] 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 *ICML*. PMLR, 2021.

[14] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023.

[15] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021.

[16] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *CVPR*, 2023.

[17] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023.

[18] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[19] Mukund Varma, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that nerf needs? In *ICLR*, 2023.

[20] Jiteng Mu, Shen Sang, Nuno Vasconcelos, and Xiaolong Wang. Actorsnerf: Animatable few-shot human rendering with generalizable nerfs. In *ICCV*, pages 18391–18401, 2023.

[21] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. Advances in neural rendering. In *arXiv:2111.05849*, 2021.

[22] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *CVPR*, 2022.

[23] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *NeurIPS*, 2022.

[24] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *International Conference on 3D Vision (3DV)*, 2022.

[25] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[28] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[29] 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 *CoRL*, 2023.

[30] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *CVPR*, 2022.

[31] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. In *SCIENCE ROBOTICS*, 2023.

[32] Nur Muhammad (Mahi) Shafiullah, Chris Paxton, Lerrel Pinto1 Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. In *RSS*, 2023.

[33] 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 *ICRA*, 2023.

[34] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *IROS*, 2022.

[35] Dhruv Shah, Michael Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning (CoRL)*, 2023.

[36] Saeid Amiri, Kishan Chandan, and Shiqi Zhang. Reasoning with scene graphs for robot planning under partial observability. *IEEE Robotics and Automation Letters (RAL)*, 7(2):5560–5567, 2022.

[37] Zachary Ravichandran, Lisa Peng, Nathan Hughes, J Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9272–9279, 2022.

[38] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *In Neural Information Processing Systems (NeurIPS)*, 2020.

[39] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. In *arXiv:2303.07798*, 2023.

[40] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *CVPR*, 2022.

[41] Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Topological semantic graph memory for image-goal navigation. In *CoRL*, 2022.

[42] Pierre Marza, Laetitia Matignon, Olivier Simonin, and Christian Wolf. Multi-object navigation with dynamically learned neural implicit representations. In *ICCV*, 2023.

[43] Leslie Pack Kaelbling and Tomas Lozano-Perez. Unifying perception, estimation and action for mobile manipulation via belief space planning. In *ICRA*, 2012.

[44] Charles Sun, Jędrzej Orbik, Brian Yang Coline Devin, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning with applications to mobile manipulation. In *CoRL*, 2021.

[45] Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Mart´ın-Mart´ın. Error-aware imitation learning from teleoperation data for mobile manipulation. In *CoRL*, 2021.

[46] Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. In *ICLR*, 2023.

[47] Priyam Parashar, Vidhi Jain, Xiaohan Zhang, Jay Vakil1, Sam Powers, Yonathan Bisk, and Chris Paxton. Slap: Spatial-language attention policies. In *CoRL*, 2023.

[48] Fei Xia, Chengshu Li, Roberto Martın-Martın, Or Litany, Alexander Toshev, and Silvio Savarese. Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation. In *ICRA*, 2021.

[49] Naoki Yokoyama, Alex Clegg, Joanne Truong, Eric Undersander, Tsung-Yen Yang, Sergio Arnaud, Sehoon Ha, Dhruv Batra, and Akshara Rai. Asc: Adaptive skill coordination for robotic mobile manipulation. In *arXiv:2304.00410*, 2023.

[50] Xiaoyu Huang, Dhruv Batra, Akshara Rai, and Andrew Szot. Skill transformer: A monolithic policy for mobile manipulation. In *ICCV*, 2023.

[51] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language model. In *CoRL*, 2023.

[52] Kenneth Blomqvist, Michel Breyer, Andrei Cramariuc, Julian Forster, Margarita Grinvald, Florian Tschopp, Jen Jen Chung, Lionel Ott, Juan Nieto, and Roland Siegwart. Go fetch: Mobile manipulation in unstructured environments. In *arXiv:2004.00899*, 2020.

[53] Simon Zimmermann, Roi Poranne, and Stelian Coros. Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm. In *ICRA*, 2021.

[54] Riccardo Parosi, Mattia Risiglione, Darwin G. Caldwell, Claudio Semini, and Victor Barasuol. Kinematically-decoupled impedance control for fast object visual servoing and grasping on quadruped manipulators. In *IROS*, 2023.

[55] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *CVPR*, 2023.

[56] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[57] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *ICRA*, 2023.

[58] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[59] 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. *arXiv preprint arXiv:2304.02643*, 2023.

[60] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[61] Yang Fu, Shalini De Mello, Xueting Li, Amey Kulkarni, Jan Kautz, Xiaolong Wang, and Sifei Liu. 3d reconstruction with generalizable neural fields using scene priors. *arXiv preprint arXiv:2309.15164*, 2023.

[62] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *ICCV*, 2023.

[63] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022.

[64] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *RSS*, 2022.

[65] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*. PMLR, 2020.

[66] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, 2021.

[67] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021.

[68] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[70] Otto Seiskari, Pekka Rantalankila, Juho Kannala, Jerry Ylilammi, Esa Rahtu, and Arno Solin. Hybvio: Pushing the limits of real-time visual-inertial odometry. In *WACV*, 2022.

[71] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. https://ompl.kavrakilab.org.

[72] https://gazebosim.org/home.

[73] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.