Learning Generalizable Feature Fields for Mobile Manipulation

Ri-Zhao Qiu*[1], Yafei Hu*[1,2], Yuchen Song*[1], Ge Yang[3], 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
https://geff-b1.github.io

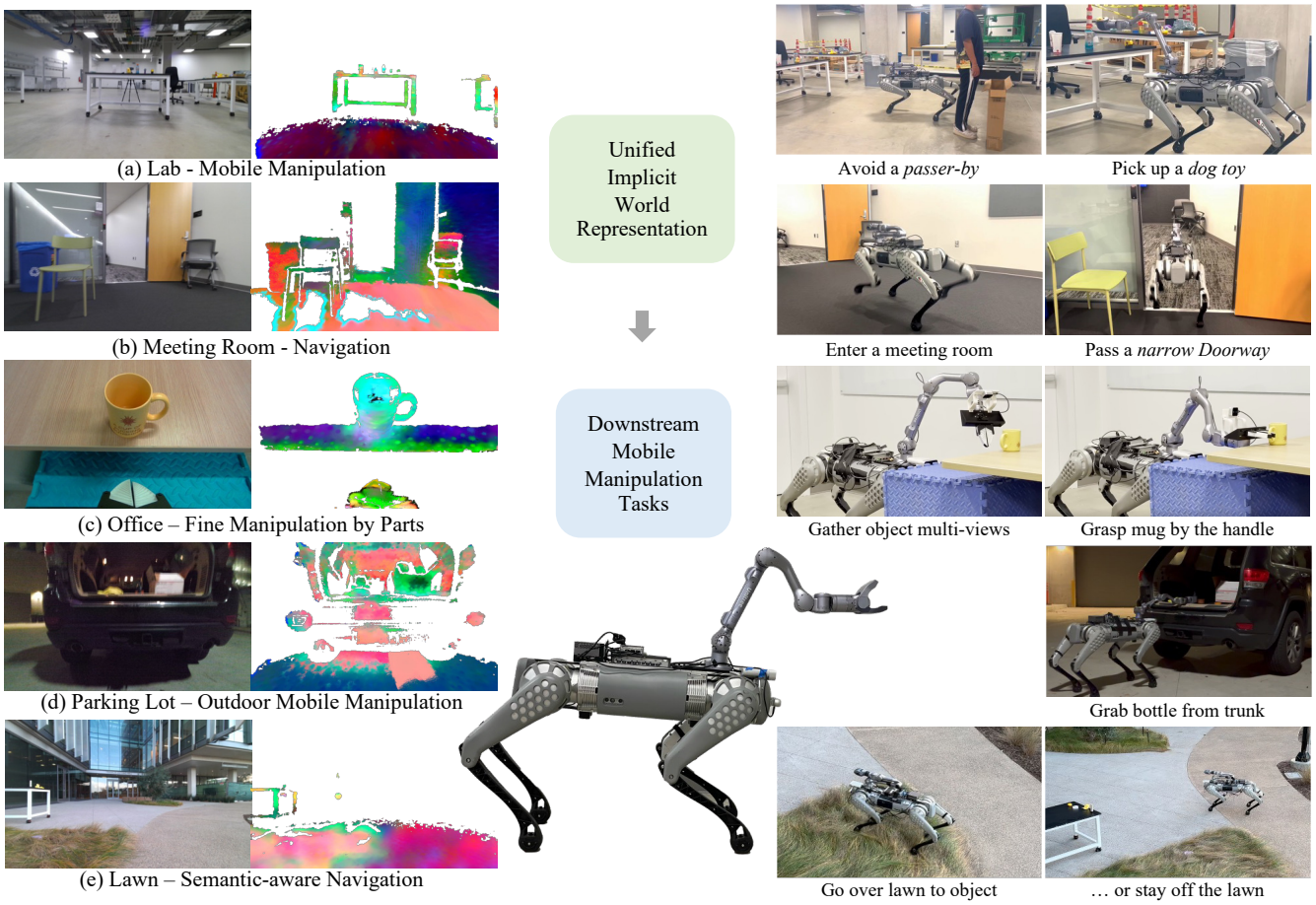


Fig. 1: **GeFF**, **Ge**neralizable **F**eature **F**ields, provide unified implicit scene representations for both robot navigation and manipulation in real-time. We demonstrate the efficacy of GeFF on **open-world mobile manipulation**, **semantic-aware navigation**, and **zero-shot manipulation by parts** under diverse scenes ((a) work in a lab where a person walks in, (b) enter a meeting room with narrow entrance, (c) fine part-level manipulation, (d) grasp objects in a parking lot, and (e) semantic-aware navigation near a lawn). The visualization of the feature fields is obtained by PCA of rendered features. For best illustration, please check out the supplementary video.

***Abstract*— An open problem in mobile manipulation is *how to represent objects and scenes* in a unified manner so that robots can use both for navigation and manipulation. The latter requires capturing intricate geometry while understanding fine-grained semantics, whereas the former involves capturing the complexity inherent at 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 quantitatively evaluate GeFF's ability for open-vocabulary object-/part-level manipulation and show that GeFF outperforms point-based baselines in runtime and storage-accuracy trade-offs, with qualitative examples of semantics-aware navigation and articulated object manipulation.**

I. Introduction

Building a personal robot that can assist with common chores has been a long-standing goal of robotics [12, 25,

50]. This paper studies the task of open-vocabulary mobile manipulation, where a robot needs to navigate through diverse scenes and manipulate objects based on language instructions. This task, while seemingly easy for humans, remains challenging for autonomous robots. 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 robots, there does not exist a unified scene representation that captures geometry and semantics for navigation and manipulation tasks.

Recent approaches in navigation seek representations such as geometric maps (with semantic labels) [1, 32, 45] and topological maps [38, 39] to handle large-scale scenes, but are not well integrated with manipulation requirements. Manipulation, on the other hand, often relies on dense scene representation such as implicit surfaces or meshes [34, 48, 58] to compute precise grasping poses, which are not typically encoded in navigation representations. More importantly, supporting semantics-aware navigation with open-vocabulary object queries requires grounding to **geometric and semantic concepts** in the environment. The lack of a unified representation leads to unsatisfactory performance in open-vocabulary manipulation in large scenes [53]. Performing coherent open-vocabulary perception for both navigation and manipulation remains a significant challenge.

We present a novel ***scene-level*** **Ge**neralizable **F**eature **F**ield (**GeFF**) as a ***unified*** representation for navigation and manipulation, trained with neural rendering akin to Neural Radiance Fields (NeRFs) [26]. Instead of fitting a single static NeRF, GeFF only requires a single feed-forward pass to update the scene representation during inference. As a unified representation, GeFF stands out with two more advantages: (i) GeFF can decode multiple 3D scene representations from a posed RGB-D stream, including signed distance function (SDF) and point cloud, and (ii) performing feature distillation from a pre-trained Vision-Language Model (VLM), *e.g.*, CLIP [33], GeFF provides language-conditioned semantics. Thus, GeFF mitigates the aforementioned discrepancy by supporting both real-time semantics-aware navigation (*e.g.,* avoiding humans) and zero-shot object part manipulation (*e.g.,* grasping mugs and tools by handles).

Using a quadrupedal mobile manipulator, we demonstrate that GeFF enables capabilities such as object-/part-level manipulation, semantics-aware navigation, and the potential to support articulated manipulation. We quantitatively show that GeFF outperforms existing point-based [11] and implicit [18] methods in open-vocabulary scene representation for mobile manipulation. Notably, the overall success rate **outperforms the best baseline by 19.2 absolute points on averaged object-level and part-level manipulation, while maintaining real-time efficiency**. In addition, we also qualitatively show that GeFF can be used to provide perception for other tasks such as semantics-aware navigation and articulated manipulation. We plan to release the pre-trained models and the source code.

II. Related Work

Generalizable NeRFs. Generalizable NeRFs extend conventional NeRFs' ability to render detailed novel views to scenes that come with just one or two images [27, 47, 49, 52, 56]. They replace the time-consuming per-scene optimization with a single feed-forward process through a network. Existing work [35, 44] 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 representation generation on mobile robots.

Feature Distillation in NeRF. Beyond just synthesizing novel views, recent work [18, 20, 46, 52] attempted to combine NeRF with feature distillation [3, 28, 33, 36] to empower neural fields with semantic understanding of objects [20, 46, 52], scenes [18, 40] and downstream robotic applications [40, 57]. PartSLIP [21] and FeatureNerf [52] performs part-level segmentation of objects, but require complete point clouds. Most closely related to our work, LERF-TOGO [18, 34] and F3RM [40] distill CLIP features for tabletop manipulation. We show that the conditional CLIP queries proposed in LERF-TOGO [34] apply to GeFF for part-based manipulation as well. Nonetheless, previous work cannot be easily adapted for mobile manipulation due to the expensive per-scene optimization scheme [18, 20] or restrictions to object-level representations [52]. In contrast, GeFF runs real-time on mobile robots.

Mobile Manipulation. Besides work that perform closed-set mobile grasping [10, 16, 30, 31, 41, 43, 51, 54, 60], there have been some recent work [5, 11, 14, 17, 22, 24, 55] that leverage 2D foundation vision models to for open-vocabulary mobile grasping and demonstration-based mobile manipulation [2]. Existing open-vocabulary manipulation methods project predictions from large-scale models [19, 33] directly onto explicit representations. This may require (1) offline optimization [11], expensive storage costs allowing only room-scale scenes and object-level grasping [11, 22]. GeFF, on the other hand, builds a *latent and unified representation* for larger-scale outdoor environments and part-level grasping in real-time.

III. GeFF for Mobile Manipulation

A. Problem Statement

Given a coordinate $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d}$ on the unit sphere $\mathbf{S}^2$, NeRF [26] adopts an occupancy mapping $\sigma_\theta(\mathbf{x}) : \mathbb{R}^3 \to [0, 1]$ and a color mapping $\mathbf{c}_\omega(\mathbf{x}, \mathbf{d}) : \mathbb{R}^3 \times \mathbf{S}^2 \to \mathbb{R}^3$. Consider a ray $\mathbf{r}$ from a camera viewport with origin $\mathbf{o}$ and direction $\mathbf{d}$. 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 bounding distances, $T(t) = \exp(-\int_{t_n}^{t} \sigma_\theta(s)\mathrm{d}s)$ is the transmittance capturing cumulative occupancy, and $\alpha_\theta(r(t))$ is the opacity value at $r(t)$ (in NeRF [26], $\alpha_\theta = \sigma_\theta$).

Let Ω be the space of RGB-D images. Consider N posed RGB-D frames $\mathcal{D} = \{(F_i, \mathbf{T}_i)\}_{i=1}^N$, $F_i \in \Omega$, $\mathbf{T}_i \in$

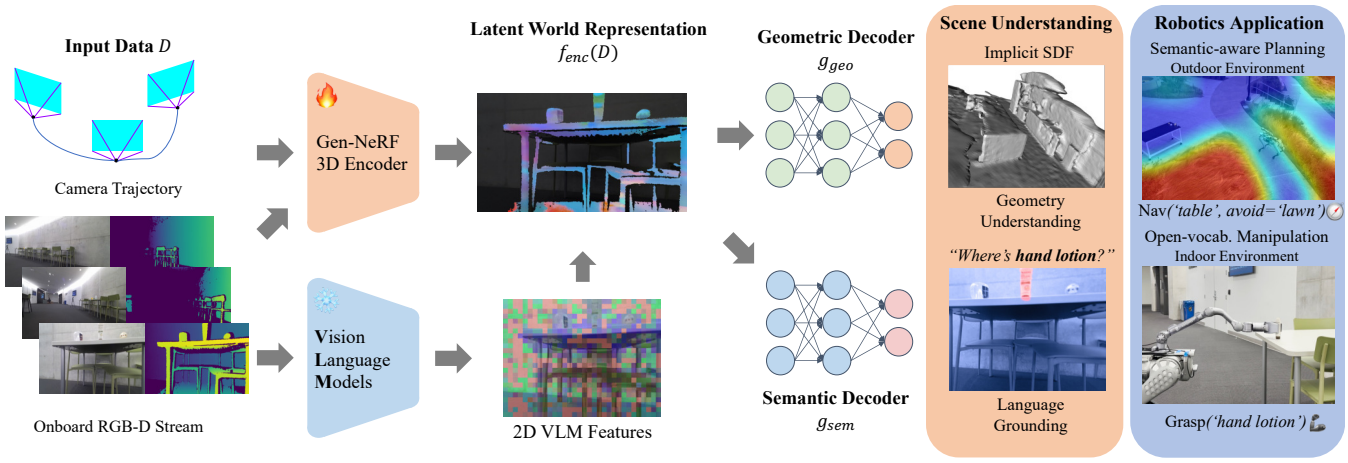


Fig. 2: Pre-trained as a generalizable NeRF encoder, **GeFF** provides a unified scene representation to support robot tasks from a onboard RGB-D stream, offering both real-time geometric information for planning and language-grounded semantics query capability. Compared to LERF [18], GeFF runs in real-time without costly per-scene optimization, which enables many potential robotics applications. We demonstrate the efficacy of GeFF in **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.

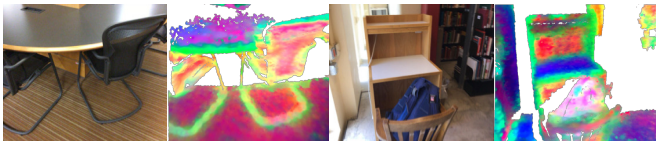


Fig. 3: **Generalizable NeRFs acquire geometric and semantic priors:** RGB images are input views from ScanNet [6], color images are PCA visualizations of feature volume projected to the input camera view encoded by an RGB-D Gen-NeRF [8] encoder. Note how semantically similar structures acquire similar features.

$\mathbf{SE}(3)$. Our goal is to create a ***unified*** scene representation that captures geometric and semantic properties for robot loco-manipulation tasks. Specifically, we aim to design an encoding function $f_{enc}(\cdot) : (\Omega \times \mathbf{SE}(3))^N \rightarrow \mathbb{R}^{N \times C}$ that compresses $\mathcal{D}$ to a latent representation, and decoding functions $g_{geo}(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^m$ and $g_{sem}(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^n$ that decode the latents into different geometric and semantic features at different positions in 3D space. The geometric and semantic features can then serve as input to a downstream planner. We aim to design these 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.*, $f_{enc}(\mathcal{D}_1 \cup \mathcal{D}_2) = f_{enc}(\mathcal{D}_1) \oplus f_{enc}(\mathcal{D}_2)$)
- **Implicit.** The encoded latents $f_{enc}(D)$ are organized in a sparse implicit representation to enable more efficient scaling to large scenes compared to storing $\mathcal{D}$.
- **Open-world.** The semantic knowledge from g_{sem} is open-set and aligned with ***language***, so the robot can perform open-world perception.

We build GeFF upon generalizable NeRFs to satisfy these requirements. An overview of our method is shown in Fig. 2.

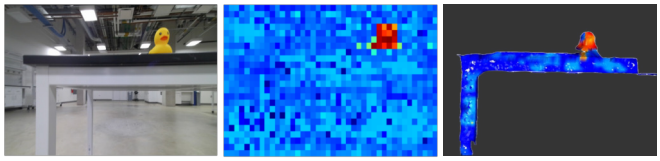
B. Learning Scene Priors via Neural Synthesis

Generalizable NeRFs (Gen-NeRFs) offer an effective pre-training objective for rich geometric and semantic priors [8, 15, 52]. Fig. 3 shows an illustration, rendering the latent feature volume from an RGB-D Gen-NeRF encoder $f_{env}(\cdot)$ [8] trained to synthesize novel views on the ScanNet [6] dataset. The colors correspond to the principal components of the latent features. We observe separations between objects and the background, despite that explicit semantic supervision was not provided during training.

GeFF uses two types of supervision to enhance these priors — semantics using 2D features and geometry using SDF.

Supervision (i): Language-Alignment via Feature Distillation. Although we have shown that Gen-NeRF encoders implicitly capture geometric and semantic cues, the representation is less useful if it is not ***aligned*** to other feature modalities, such as language. To enhance the representation capability, in GeFF we 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 contrast to previous works [18, 20, 52], which either require costly per-scene optimization [18, 20] or is limited to object-centric representation [52], 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 isotropic, as the semantics of an object should be view-independent regardless of the viewing directions. We can



(a) (b) (c)

Fig. 4: **GeFF compresses and refines multi-view observations:** (a) single RGB view; (b) coarse 2D CLIP heatmap with query 'toy duck'; (c) 3D heatmap from GeFF with clean boundary reconstructed from compressed latent representation.

render 2D features for pre-training via

$$\hat{\mathbf{F}}(r) = \int_{t_n}^{t_f} T(t)\alpha(r(t))g_{sem}(\mathbf{r}(t), f_{enc}(\mathcal{D}))\mathrm{d}t\,, \quad (2)$$

which is modified from Eq. 1. 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

$$f_{enc}(\mathcal{D}) = \textsc{ConCat}\left(\hat{f}_{enc}(\mathcal{D}), f_{teacher}(\mathcal{D})\right)\,, \quad (3)$$

where $\hat{f}_{enc}$ is a trainable encoder and $f_{teacher}$ is a pre-trained vision model with frozen weights. The final feature rendering loss is then given by standard L2 loss between $\hat{\mathbf{F}}$ and $\mathbf{F}$, where $\mathbf{F}$ is the reference feature obtained by running $f_{teacher}$ on ground-truth novel views and $\hat{\mathbf{F}}$ is the predicted feature. Note that the input views and the rendered novel views are different adjacent views.

Model for Distillation. Our proposed feature distillation method for scene-level generalizable NeRFs is model-agnostic. In this work, since we are interested in *open-vocabulary* tasks, we choose MaskCLIP [59] as $f_{teacher}$. MaskCLIP offers coarse (see Fig. 4) features but runs in real-time on mobile robots.

Supervision (ii): Depth Supervision via Neural SDF. We use a signed distance network $s(\mathbf{x}) = g_{geo}(\mathbf{x}, f_{enc}(\mathcal{D}))$ to decode feature field into metric geometry, which is based on existing work [8, 29, 49]. Doing so has two advantages over previous work [56]: 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 [29].

To provide supervision for g_{geo} during pre-training, we differentiably convert SDF values into 2D depths following iSDF [29]. The main difference with iSDF [29] is that we condition g_{geo} with $f_{enc}(\mathcal{D})$, which ***does not require optimization for novel scenes***. We represent the opacity function α in Eq. 2 using $s(\mathbf{x})$

$$\alpha(r(t)) = \textsc{Max}\left(\frac{\sigma_s(s(\mathbf{x})) - \sigma_s(s(\mathbf{x}+\Delta))}{\sigma_s(s(\mathbf{x}))}, 0\right)\,, \quad (4)$$

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

$$\hat{\mathbf{D}}(r) = \int_{t_n}^{t_f} T(t)\alpha(r(t))d_i\mathrm{d}t\,, \quad (5)$$

where d_i is the distance from the current ray marching position to the camera origin. Similar to Eq. 2, the rendered depth can be supervised via standard L2 loss.

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

$$\mathcal{L} = \lambda_1\mathcal{L}_{col} + \lambda_2\mathcal{L}_{depth} + \lambda_3\mathcal{L}_{sdf} + \lambda_4\mathcal{L}_{eik} + \lambda_5\mathcal{L}_{feat}\,, \quad (6)$$

where λ_is are empirically insensitive hyperparameters used to balance loss scales; $\mathcal{L}_{col}$, $\mathcal{L}_{depth}$, and $\mathcal{L}_{feat}$ are rendering loss for Eq. 1, Eq. 5, and Eq. 2. The SDF loss $\mathcal{L}_{\text{sdf}}$ and Eikonal regularization loss [9] $\mathcal{L}_{\text{eik}}$ are standard losses used by existing methods to ensure smooth SDF values.

C. Implementing Open-Vocabulary Mobile Manipulation

Scene Mapping with GeFF. GeFF encodes posed RGB-D frames to a latent 3D volume represented as a sparse latent point cloud, which can be built by concatenating per-frame observations. The camera poses are provided by an off-the-shelf VIO method [37].

Decoded Representations. Though GeFF supports continuous decoding, it is inefficient to generate all possible representations densely on-the-fly. For this work, we decode the latent representation into discretized point clouds as geometric representations for navigation and manipulation. We then compute 2D grid by projecting the decoded 3D points and compute features for each grid cell by averaging the features of related points. This enhances basic units (*i.e.,* points and grid cells) with features from g_{sem}.

Handling Language Query. Following standard protocols [33], GeFF takes in positive text queries and negative text queries (*e.g., ceiling*). To rate the language similarity, we compared decoded point features with text features using cosine similarity with a temperatured softmax. We sum up the probabilities of positive queries as the similarity score. For part-level language query, we use the conditional CLIP query technique proposed by Rashid *et al.* [34]. After the initial object is segmented, conditional CLIP query performs another pass of language query *conditioned on the segmented object with part-level prompt* for part segmentation.

GeFF for Navigation. We consider the navigation of the quadruped robot as a 2D navigation problem following existing work [4, 53, 55]. Given text queries, we compare text embedding to grid embeddings. We use DBSCAN [7] to cluster high-response points for goal location and assign semantic affordances to grid cells. With an affordance-aware A* planner, this achieves semantic-aware navigation. Note that the 2D occupancy map is updated in real-time.

GeFF for Object-level Manipulation. After the robot arrives at the goal receptacle, it searches for the target object by comparing semantics in points with given text, and uses DBSCAN to represent the target object as a centroid. In practice, we found that the parallel gripper has a high success

TABLE I: Open-vocabulary mobile manipulation. Navigation success (Nav. Succ.) and composite mobile manipulation success (Mobile. Mani. Succ.) are reported for object-level tasks. For part-level tasks, we report manipulation success rates with different object-part queries (*e.g.,* mug-handle: grasping various mugs by handles). Latency reports the delay from reception of frames to decoded semantic features on the onboard AGX Orin. The overall success is the average of object-level and part-level manipulation. ⋆ methods require offline optimization with all observations batched together.

		Object-level Mobile Manipulation		Part-level Manipulation				
Method	Latency	Nav. Succ.	Mobile Mani. Succ.	Mug-Handle	Tool-grip	Cart-bar	Avg. Succ.	Overall Succ.
GeFF (Ours)	0.39s	**94.4%**	61.1%	**44.4%**	**66.7%**	**80.0%**	**63.7%**	**62.4%**
LERF⋆ [18]	∼2 hrs⋆	72.2%	44.4%	36.1%	20.0%	70.0%	42.0%	43.2%
ConceptGraph⋆ [11]	∼200s⋆	**94.4%**	**72.2%**	0%	20%	15%	11.6%	41.9%
ConceptGraph-Online	4.63s	5.56%	5.56%	0%	0%	15%	5%	5.3%

TABLE II: **Ablation of auxiliary CLIP input** (Eq. 3) on object-level mobile manipulation in diverse scenes. Navigation success rates (Navi.) and composite mobile manipulation success rates (Mani.) are reported.

	Meeting Room		Kitchen		
Methods	Navi.	Mani.	Navi.	Mani.	Overall
GeFF (Ours)	**13/15**	**8/15**	**12/18**	**8/18**	**41/66**
GeFF (no aux)	9/15	5/15	7/18	4/18	25/66
LERF [18]	6/15	3/15	8/18	5/18	22/66

TABLE III: Mobile manipulation under **scene change**, where objects are added after the initial scan. Note that methods [11, 18] with expensive training requirement do not handle scene change.

Method	Change	Lab	Meet. Rm.	Kitchen
GeFF	✗	7/9	7/9	8/9
	✓	4/9	6/9	8/9
LERF [18]	✗	6/9	7/9	4/9
	✓	NA*	NA*	NA*

rate in object-level grasping via an intuitive open-push-close gripper action sequence with trajectories computed by a sample-based planner (OMPL planner [42]).

GeFF for Part-level Manipulation. For objects that involve intricate geometry (*e.g.*, mug/tool with handles), it is counter-intuitive to solve the grasping problem with a centroid. In such cases, the user can provide specific parts to grasp via language. In GeFF, after the object centroid is localized, the robot can optionally use its in-wrist camera to gather multiple views, which adds millimeter-scale details to the representation. We then perform conditional CLIP queries and DBSCAN using significantly smaller EPS (*e.g.,* 1cm) to determine grasping location.

IV. Experiments

A. Experimental Setup

Training Details. GeFF is pre-trained on the ScanNet dataset [6]. for 50 epochs with 8 RTX3090 GPUs in 6 days. We use the ViT-L CLIP model as $f_{teacher}$. Encoders are implemented as a mixture of PointNet and ResNet [13] following Fu et al. [8]. Decoders are implemented as MLPs.

Robot Platforms. We use the Unitree B1 as the base robot with a Unitree Z1 arm mounted on top of it. Besides a stereo camera and a structured light camera mounted at the robot head, the part-level experiments also uses an in-wrist camera to gather multi-view images. (See website for visuals).

Real-world Evaluation. For quantitative experiments, we use 4 environments: a $25m^2$ lab , a $30m^2$ meeting room, a $60m^2$ community kitchen, and a $15m^2$ office. For object-level experiments, unless otherwise noted, we use a total of 17 objects (6 misc., 5 office items, and 6 culinary items) including 8 novel categories that GeFF had ***not seen*** during pre-training. For part-level manipulation, we use three different object categories with 4 instances each.

Experiment Protocol. For all settings, we first manually drive the robot to build an initial representation of the scene to perceive receptacles (replaceable by standard robotic exploration algorithms). Then we provide task-related receptacle and object names to the robot.

Baseline Implementation. We choose two recent open-vocabulary scene representations as baselines. ConceptGraphs [11] is a state-of-the-art open-vocabulary scene-level representation. Similar to OK-Robot [22], it uses pre-trained vision models [19, 23] for perception. Since both **ConceptGraph**⋆ and **LERF**⋆ require offline batch processing of all images, we process observed frames on a desktop computer. After which we manually provide object goals. **ConceptGraph-Online** is an online variant of CG, where it drops incoming frames if the previous frame is not finished processing. Since CG does not run on the AGX Orin, we reuse the same pipeline of ConceptGraph⋆ but downsample the frames to match the latency. All representations are constructed by poses estimated by onboard VIO.

B. Evaluation

We answer important **R**esearch **Q**uestions: How is GeFF compared to other open-vocabulary scene representation methods (A1, A2, A3, A4)? How is GeFF compared to simple projection baseline (A6)? What were the design choices (A5)? Can GeFF be used for diverse tasks (A7)?

A1. ConceptGraph requires offline optimization and breaks when real-time requirement is enforced. From Tab. I, we can see that ConceptGraph works at the cost of expensive offline processing, which is not suitable for mobile robots. When ConceptGraph is granted offline processing using desktop-level compute, it achieves *slightly* better results than GeFF on object-level grasping. However, when it is forced to perform online inference, we empirically

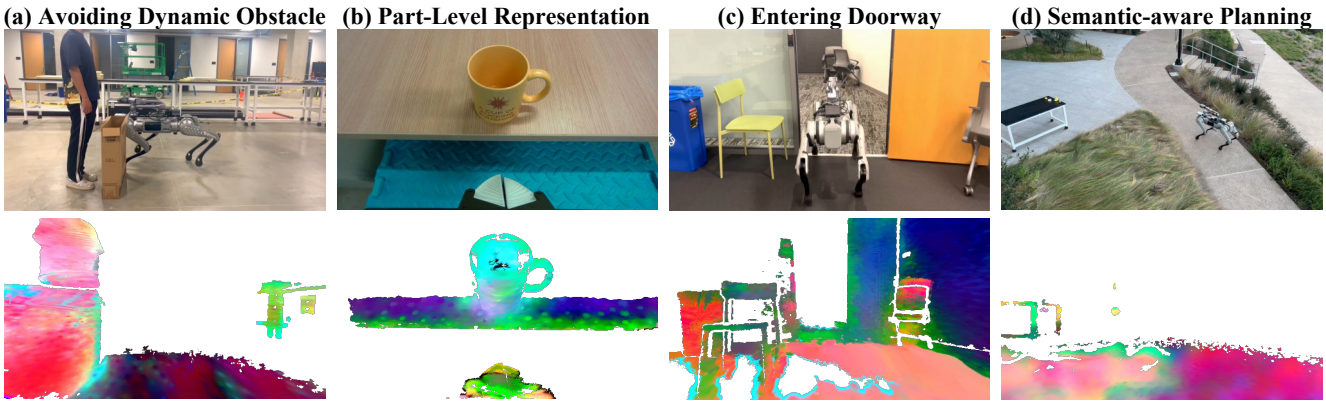


Fig. 5: **Qualitative results of GeFF for diverse tasks:** (a) real-time update for dynamic person detection; (b) GeFF enables manipulation by parts; (c) entering a narrow doorway; (d) semantics-aware planning with affordance of *'lawns'*. The results are animated in the supplementary video. Images in the second row are PCA visualization of first-person GeFF features.

TABLE IV: GeFF learns geometric priors to reconstruct geometry from compressed latent representation. Both GeFF and projection baselines downsample the depth and MaskCLIP features to at most 512 points. Depths are reconstrcuted/upsampled and compared to reference depth.

Method	Depth L2 Error↓
GeFF	**0.012**
Projection (Nearest interpolation)	0.061
Projection (Bilinear interpolation)	0.040

observe its internal point cloud merging design breaks due to its assumption of adjacent frame proximity, which leads to degenerate representations and bad success rate.

A2. ConceptGraph fails to respond to part-level queries. Specifically designed for object-level representations, ConceptGraph can not support part-level grasping (*e.g.,* grasping a screwdriver by handle instead of shank), which is evident from Tab. I. Specifically, it generates no or bad responses to part-level queries such as *handles* or *grips*, which is due to lack of part-level training data in the open-vocabulary detector [23] that ConceptGraph relies on.

A3. Unlike GeFF, LERF requires offline processing and does not provide clear boundary. LERF [18], another feature field method, is an RGB-only method with view-dependent features. Thus we select the point with maximum responses in features rendered from training views as the goal location. Due to lack of geometric supervision, *LERF often fails due to (1) noisy responses from under-observed areas and (2) unclear object boundaries*. However, as a continuous implicit method, **LERF show significantly better performance on part-level manipulation than ConceptGraph**, which is consistent with our finding that continuous representation is better suited for part-level representation.

A4. GeFF works when scene changes with slightly worse performance. For manipulation under **scene change**, we place a subset of objects (hand lotion, bottle, dog toy) on the table *after* the initial scan with 3 trials each. Tab. III shows the results. Both LERF [18] and CG [11] are not applicable for scene changes as they require costly re-training. One potential cause for the decrease is the lack of multi-view observations as the robot only gets a front view when it approaches the receptacle.

A5. Auxiliary 2D input helps with generalization. We ablate GeFF the effectiveness of Eq. 3 in more diverse environments in Tab. II. Specifically, we found that, if auxiliary input is not used, GeFF shows decreased performance especially on objects absent from pre-training on ScanNet [6]. We believe that auxiliary input provides a 'shortcut' generalization beyond training data, which may replaced by a significantly larger training scale.

A6. The learned geometric priors are effective at compression. To evaluate the learned geometric priors, we reconstruct depth from the latent representation to compare it with reference depth. For a given RGBD frame, GeFF encodes it to 512 latent points and reconstructs the depth. The simple projection baseline downsamples the given RGBD frame to 512 pixels, and interpolates back to the original resolution. The resulting L2 errors between reconstructed depths and reference depths are given in Tab. IV using 10 validation scenes of the ScanNet dataset, where GeFF shows significantly better geometric error.

A7. GeFF can serve as the 3D perception backbone for diverse tasks. We show qualitatively in both Fig. 5 and the supplementary material that GeFF features are fine-grained and real-time enough to perform diverse tasks beyond grasping, such as **dynamic obstacle avoidance, semantic-aware navigation, and articulated manipulation for door opening**, which highlights its potential to provide 3D representation for robotics tasks.

V. 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 quadruped 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 open-set mobile manipulation tasks.

References

[1] A. Asgharivaskasi and N. Atanasov, "Semantic OctTree Mapping and Shannon Mutual Information Computation for Robot Exploration," *IEEE Transactions on Robotics (T-RO)*, vol. 39, no. 3, pp. 1910–1928, 2023.

[2] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation," *arXiv preprint arXiv:2405.01527*, 2024.

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *International Conference on Computer Vision (ICCV)*, 2021.

[4] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *In Neural Information Processing Systems (NeurIPS)*, 2020.

[5] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," in *ICRA*, 2023.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017.

[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, 1996.

[8] Y. Fu, S. De Mello, X. Li, A. Kulkarni, J. Kautz, X. Wang, and S. Liu, "3d reconstruction with generalizable neural fields using scene priors," *arXiv preprint arXiv:2309.15164*, 2023.

[9] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *ICML*, PMLR, 2020.

[10] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill mobile manipulation for object rearrangement," in *ICLR*, 2023.

[11] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.

[12] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9094–9104, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[14] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *ICRA*, 2023.

[15] D. Huang, S. Peng, T. He, H. Yang, X. Zhou, and W. Ouyang, "Ponder: Point cloud pre-training via neural rendering," in *ICCV*, 2023.

[16] X. Huang, D. Batra, A. Rai, and A. Szot, "Skill transformer: A monolithic policy for mobile manipulation," in *ICCV*, 2023.

[17] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[18] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *ICCV*, 2023.

[19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[20] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," *NeurIPS*, 2022.

[21] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su, "Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models," in *CVPR*, 2023.

[22] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024.

[23] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[24] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *arXiv preprint arXiv:2404.13696*, 2024.

[25] J. M. Marques, J.-C. Peng, P. Naughton, Y. Zhu, J. S. Nam, and K. Hauser, "Commodity telepresence with team avatrina's nursebot in the ana avatar xprize finals," in *ICRA 2023 2nd Workshop on Toward Robot Avatars*, 2023.

[26] 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 *ECCV*, 2020.

[27] J. Mu, S. Sang, N. Vasconcelos, and X. Wang, "Actorsnerf: Animatable few-shot human rendering with generalizable nerfs," in *ICCV*, 2023, pp. 18 391–18 401.

[28] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[29] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "Isdf: Real-time neural

signed distance fields for robot perception," in *RSS*, 2022.

[30] P. Parashar, V. Jain, X. Zhang, J. Vakil1, S. Powers, Y. Bisk, and C. Paxton, "Slap: Spatial-language attention policies," in *CoRL*, 2023.

[31] R. Parosi, M. Risiglione, D. G. Caldwell, C. Semini, and V. Barasuol, "Kinematically-decoupled impedance control for fast object visual servoing and grasping on quadruped manipulators," in *IROS*, 2023.

[32] R.-Z. Qiu, Y. Sun, J. M. C. Marques, and K. Hauser, "Real-time semantic 3d reconstruction for high-touch surface recognition for robotic disinfection," in *IROS*, 2022.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, PMLR, 2021.

[34] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *Conference on Robot Learning*, 2023.

[35] D. Rebain, M. Matthews, K. M. Yi, D. Lagun, and A. Tagliasacchi, "Lolnerf: Learn from one look," in *CVPR*, 2022.

[36] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[37] O. Seiskari, P. Rantalankila, J. Kannala, J. Ylilammi, E. Rahtu, and A. Solin, "Hybvio: Pushing the limits of real-time visual-inertial odometry," in *WACV*, 2022.

[38] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *ICRA*, 2023.

[39] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," in *CORL*, 2023.

[40] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *CoRL*, 2023.

[41] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman, "Open-world object manipulation using pre-trained vision-language model," in *CoRL*, 2023.

[42] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, 2012.

[43] C. Sun, J. Orbik, B. Y. Coline Devin, A. Gupta, G. Berseth, and S. Levine, "Fully autonomous real-world reinforcement learning with applications to mobile manipulation," in *CoRL*, 2021.

[44] A. Tewari *et al.*, "Advances in neural rendering," in *arXiv:2111.05849*, 2021.

[45] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems," *IEEE Transactions on Robotics (T-RO)*, vol. 38, no. 4, pp. 2022–2038, 2022.

[46] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations," in *International Conference on 3D Vision (3DV)*, 2022.

[47] M. Varma, P. Wang, X. Chen, T. Chen, S. Venugopalan, and Z. Wang, "Is attention all that nerf needs?" In *ICLR*, 2023.

[48] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *ICRA*, 2019.

[49] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021.

[50] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[51] F. Xia, C. Li, R. Martın-Martın, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *ICRA*, 2021.

[52] J. Ye, N. Wang, and X. Wang, "Featurenerf: Learning generalizable nerfs by distilling foundation models," in *ICCV*, 2023.

[53] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023.

[54] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai, "Asc: Adaptive skill coordination for robotic mobile manipulation," in *arXiv:2304.00410*, 2023.

[55] N. H. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *ICRA*, 2024.

[56] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "Pixelnerf: Neural radiance fields from one or few images," in *CVPR*, 2021.

[57] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in *CoRL*, 2023.

[58] J. Zhang, N. Gireesh, J. Wang, X. Fang, C. Xu, W. Chen, L. Dai, and H. Wang, "Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion," *arXiv preprint arXiv:2309.15459*, 2023.

[59] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from clip," in *ECCV*, 2022.

[60] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm," in *ICRA*, 2021.