2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Lifelong LERF: Local 3D Semantic Inventory Monitoring Using FogROS2

Adam Rashid[1]*, Chung Min Kim[1]*, Justin Kerr[1]*, Letian Fu[1], Kush Hari[1], Ayah Ahmad[1], Kaiyuan Chen[2], Huang Huang[1], Marcus Gualtieri[3], Michael Wang[3], Christian Juette[3], Nan Tian[3], Liu Ren[3], Ken Goldberg[1]

***Abstract*— Inventory monitoring in homes, factories, and retail stores relies on maintaining data despite objects being swapped, added, removed, or moved. We introduce Lifelong LERF, a method that allows a mobile robot with minimal compute to jointly optimize a dense language and geometric representation of its surroundings. Lifelong LERF maintains this representation over time by detecting semantic changes and selectively updating these regions of the environment, avoiding the need to exhaustively remap. Human users can query inventory by providing natural language queries and receiving a 3D heatmap of potential object locations. To manage the computational load, we use Fog-ROS2, a cloud robotics platform, to offload resource-intensive tasks. Lifelong LERF obtains poses from a monocular RGBD SLAM backend, and uses these poses to progressively optimize a Language Embedded Radiance Field (LERF) for semantic monitoring. Experiments with 3-5 objects arranged on a tabletop and a Turtlebot with a RealSense camera suggest that Lifelong LERF can persistently adapt to changes in objects with up to 91% accuracy.**

## I. Introduction

The ability to monitor inventory over time and respond to natural language queries such as "where did I leave my keys?" or "where is the yellow electric screwdriver?" can be useful in homes, retail shops, repair shops, hospitals, and factories. Indoor scenes evolve over time, with objects moving, changing, appearing, or disappearing. To remain useful, systems must keep track of the current environment state. We consider how a mobile robot (wheeled or legged) can navigate using Simultaneous Localization and Mapping (SLAM) [1, 2, 3] to collect images, build, query, and update semantic 3D models over time.

Lifelong LERF enables a mobile robot with limited onboard compute to *create* a dense semantic representation that supports natural language queries, and *maintain* this representation by automatically detecting semantic differences in 3D as it navigates, updating the reconstruction to reflect these changes. Lifelong LERF uses DROID-SLAM as its monocular pose estimation backend, and uses these poses to progressively optimize a Language Embedded Radiance Field [4] (LERF) which densely embeds CLIP [5] embeddings within a Neural Radiance Field (NeRF) [6].

Prior work Evo-NeRF [7] demonstrates NeRFs can update a scene by discarding stale images and replacing them with fresh images, similar to how many SLAM models can update an occupancy distribution. However, for inventory tracking it is impractical to throw away old images and remap the entire

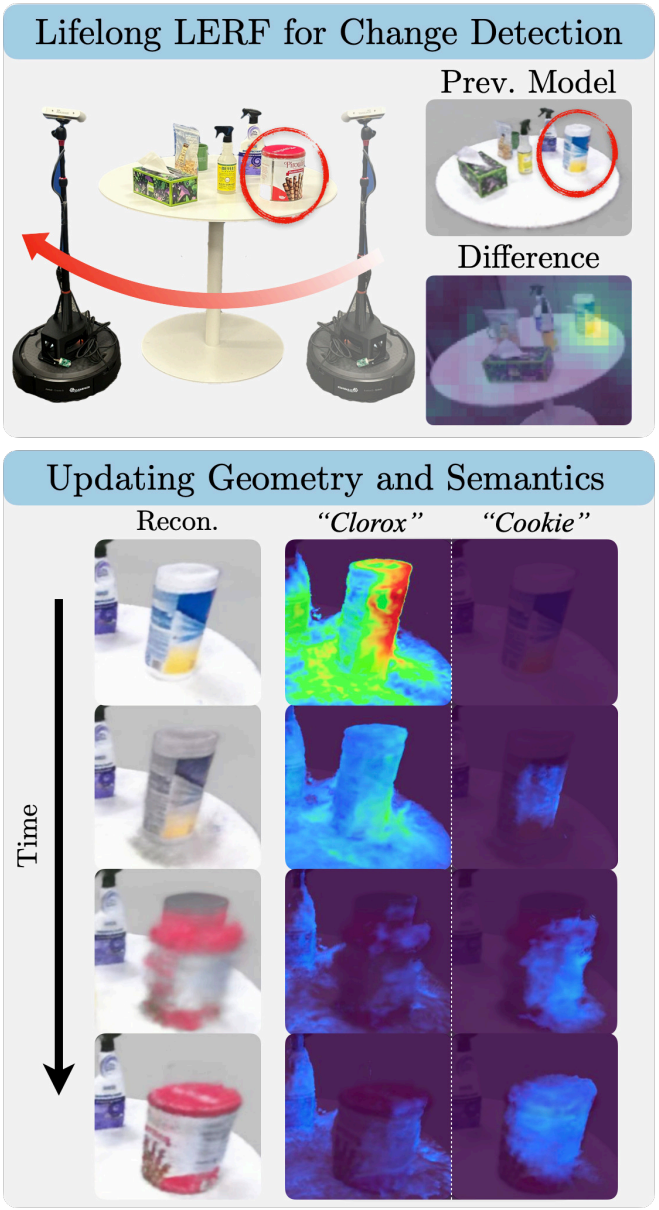


Fig. 1: *Lifelong LERF Example*. **Top**: A mobile robot takes a scan of the scene and builds a Language Embedded Radiance Field (LERF). Then, the scene is altered, for example the "Clorox" wipes are replaced by a cookie can. The robot periodically rescans the scene and identifies what has changed by rendering semantic features stored in LERF and comparing them against those extracted from the newly captured images. **Bottom**: Lifelong LERF efficiently updates the LERF using new images of the scene, progressively changing local geometry and semantics.

*Equal contribution, [1]The AUTOLab at [2]UC Berkeley, [3]Bosch

environment. In this work, we present a new method for localizing changed regions in an environment to selectively update.

Lifelong LERF detects 3D differences based on the deviation between natural language embeddings computed over a freshly captured image and a view previously rendered from the LERF at the same pose, leveraging the novel view synthesis capabilities of LERF to generate pixel-aligned difference heatmaps. Given these scene changes, the robot can update the LERF by masking out stale regions of previous input images and ignoring these pixels during future ray sampling. As fresh images are collected to the updated scene, the underlying LERF evolves to match new observations.

The compute required for running SLAM, optimizing a LERF, and computing semantic differences is too demanding for on-board robot hardware. We address this issue with cloud computing by using FogROS2 [8]. We conduct experiments on a Turtlebot 4 and evaluate Lifelong LERF's ability to maintain an updated language representation as changes are made to local tabletop scenes. We find that experiments in a tabletop environment suggest Lifelong LERF can persistently localize objects through scene changes up to 91% of the time, with the proposed semantic difference method outperforming a depth baseline in robustness to false positives and object swapping. This method could be complementary to methods using semantics for downstream applications such as language-conditioned grasping [9].

This paper contributes:

1) A novel method for detecting changes in a scene by comparing rendered language embeddings against those calculated from captured images.
2) An approach for incorporating detected scene changes progressively into an evolving LERF.
3) A system that allows Lifelong LERF to function on a robot with lightweight compute (Turtlebot 4 with Raspberry Pi 4) by leveraging FogROS2.

## II. Related Work

### A. Semantic SLAM

SLAM algorithms allow robots to localize themselves while building a map of their surroundings. While SLAM primarily focuses on geometric reconstruction, recent variants provide impressive tracking and reconstruction abilities. Several prior systems give robots the ability to model the semantic meaning of the objects in their environment. Many works used explicit scene representations, such as volumes, point clouds, scene graphs, and truncated signed distance functions (TSDF), for semantic scene understanding. McCormac *et al.* [10] proposes an object-oriented online SLAM system to produce semantically labeled TSDF object instances reconstructions and 3D foreground masks. Object segmentation [11, 12, 13, 14, 15] and detection [16] have been leveraged on volume and points clouds for semantic scene understanding. Rosinol *et al.* [17] provides a library for semantic SLAM by constructing 3D semantic meshes from semantic hand labeled images. 3D scene graph has also been used to describe a static scene either on a given mesh [18] or on the incoming frames of observations [19, 20]. Wu *et al.* [19] incrementally fuses the semantic prediction of the current observation into a global semantic graph. Hughes *et al.* [20] constructs a 3D scene graph incrementally based on topological maps of locally built Euclidean Signed Distance Functions (ESDFs).

Recent research in open vocabulary vision models [21, 22, 23, 24, 25] leads to increasing interest in creating 3D representations that incorporate semantic features. To reconstruct a static semantic SLAM, ConceptFusion [26] combines SLAM and semantic features by projecting CLIP [5] and features into 3D, where the labels are refined by 2D unsupervised segmentation.

### B. Dynamic Scene Representation

Environments often undergo changes over time. Extensive study has been done on detecting 2D scene changes [27, 28, 29]. For 3D environments, Rosinol *et al.* [30] represents dynamic scenes with moving agents with 3D dynamic scene graphs by integrating object and human detection and pose estimation model. Looper *et al.* [31] uses the Variable Scene Graph (VSG), an augmentation of 3D scene graph, to represent semantic scene changes. The variability of VSG is estimated in a supervised way. Prior work [32, 33, 34, 35] has shown success using TSDFs in capturing long-term changes. Other works have used 2D point cloud data [36] or spatiotemporal grid [37] to learn the long-term environment changes but don't provide semantic understanding. These systems also require huge storage overhead.

Neural Radiance Fields (NeRF) [6] are attractive alternative representation for high-quality scene reconstruction from pose RGB images, with an explosion of recent work on visual quality [38, 39, 40, 41, 42, 43, 44], large-scale scenes [45, 46, 47], optimization speed [48, 49, 50, 51], dynamic scenes [52, 53, 54], combining SLAM and NeRF [55, 56, 57, 58, 59] and more.

Most similar to this work are methods which seek to adapt NeRF to a continual framework, such as Evo-NeRF [7], ICNGP [60], Yan *et al.* [61], CLNeRF [62], and Fu *et al.* [63]. However, prior works either don't consider changing scenes or naively discard stale images. Note that this goal is different from dynamic NeRF reconstruction [64, 65], which must also recover the full scene motion, and 3D inpainting [66], which must hallucinate the scene without an object. In Lifelong LERF, we leverage DROID-SLAM [3] to obtain camera poses. We then actively detect changed regions and selectively update them to avoid unnecessarily discarding images.

### C. Cloud and Fog Robotics

Cloud Robotics [67] is an emerging computational paradigm for robotics that enables robots with limited onboard computing capability to gain on-demand computing hardware resources. Exemplary cloud robotics applications include SLAM, motion planning [68] and grasp planning [69, 70]. Major Cloud service providers, such as Amazon Web

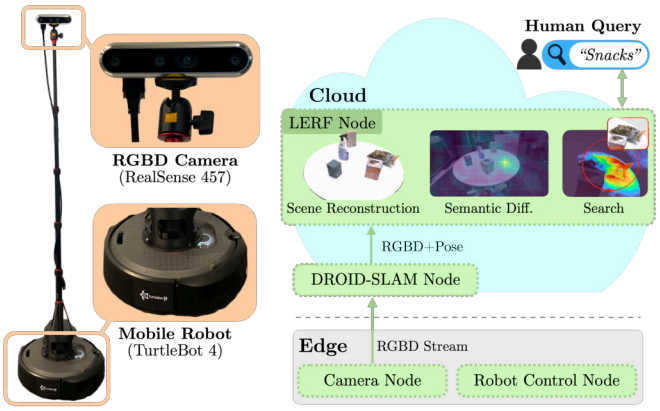


Fig. 2: *Experiment setup and FOG-ROS2 Integration.* **Left**: We use a TurtleBot 4 as the mobile robot. We mount a RealSense D457 RGBD camera on top of the robot via a monopod. **Right**: We use FOG-ROS2 to execute both DROID-SLAM and LERF on a cloud machine. In particular, after the DROID-SLAM node obtains paired RGB and depth observations, it computes the camera pose. The LERF node reconstructs the LERF and calculates if the new observation is semantically inconsistent with the stored representation.

Services (AWS) and Google Cloud Platform (GCP), provide proprietary interfaces for robotics applications to interact with cloud interfaces, such as AWS Greengrass [71] and Google Cloud Robotics Core. Rapyuta [72] is a platform for centralized management and deployment of a pipeline of robotics applications. FogROS [73] is the first open-source cloud robotics platform that offloads robotics applications to public cloud. FogROS enables robots to interact with the Cloud through the Robot Operating System (ROS) interfaces. Ichnowski *et al.* [74] present FogROS2 that supports ROS2 and more major cloud service providers. FogROS2-SGC [75, 8] secure and globally connects distributed robots with a peer-to-peer network. In this work, we apply FogROS2 to a low-powered and inexpensive mobile robot.

## III. Problem Statement

For our algorithm, we assume that:

(i) Objects are distributed on a local flat surface with sufficient visibility during a trajectory surrounding them.
(ii) During any scan of the scene, the objects in the scene are static.

We consider a typical SLAM scenario, with a wheeled, mobile robot equipped with an RGBD camera on a pole mounted on the robot. We consider several time periods: in the first time period, the robot constructs an initial model of the static environment. Between subsequent time periods, a set of objects may have been moved, added, or removed, and the goal of the robot is to detect these unknown changes and update the model accordingly. After each time period, the robot should be able to localize objects specified through natural language queries, even when objects are moved between time periods.

## IV. Method

Lifelong LERF builds a LERF while the robot moves, updates it through environment changes, and localizes natural language queries. It has three main modules:

1) Scene reconstruction takes in a stream of RGBD images and executes DROID-SLAM, concurrently building a LERF with camera poses from SLAM and RGB image observations.
2) The semantic differencing module compares incoming scene observations to the current reconstruction, producing 3D bounding boxes around changed regions.
3) The LERF updating module takes in these bounding boxes and masks stale regions of images to ensure the reconstruction adapts to the new scene.

### A. Constructing LERF

*1) Camera pose estimation:* To construct the LERF representation, we need accurate camera poses from an RGBD camera stream. To provide these, we use DROID-SLAM [3] to estimate camera poses. DROID-SLAM is a hybrid deep learning and optimization-based SLAM system that takes in monocular, stereo or RGB-D video and outputs per-keyframe disparity maps and keyframe poses. It iteratively updates predicted camera poses and pixel-wise depths through a Dense Bundle Adjustment layer, a differentiable module that computes a Gauss-Newton update to camera poses and pixel-wise depth. We choose to use DROID-SLAM for its strong pose tracking performance, and also for its ability to function without IMU and at lower camera fps (10hz), making it more robust to network latency.

We directly feed the output keyframes and poses into the LERF to use as input views. We first initialize the pose prediction process with 4 RGBD images to ensure DROID-SLAM's scene scale matches physical scale, then switch to pure RGB estimation, which we find qualitatively drifts less. After adding initial poses, camera poses are additionally refined over the course of NeRF optimization by Nerfstudio's camera optimization [45], a process which uses the differentiable rendering capabilities of NeRF to fine-tune poses, as proposed in BARF [76].

*2) Scene box determination:* Camera poses generated by DROID-SLAM are not necessarily axis-aligned with real-world coordinates, primarily because they are determined through an optimization process that aims for internal consistency rather than alignment with an external coordinate system. However, the hashgrid encoding [48] assumes all cameras fit within a pre-defined scene box, an important step for ensuring parameters are sufficiently distributed throughout the scene. To account for this, we automatically set the scene scaling based on the first time period of mapping after the whole scene has been traversed.

*3) Concurrency in LERF Computation:* Training for LERF typically takes minutes to complete, which is not sufficient in this context as the robot must actively detect environmental changes and decide whether to update the underlying LERF model. In the standard LERF training process, the input images are first pre-processed by extracting CLIP and DINO features, a process that takes minutes. To enable continual LERF optimization, we introduce two modifications to the LERF training process. First, we continuously optimize the LERF model as the robot scans

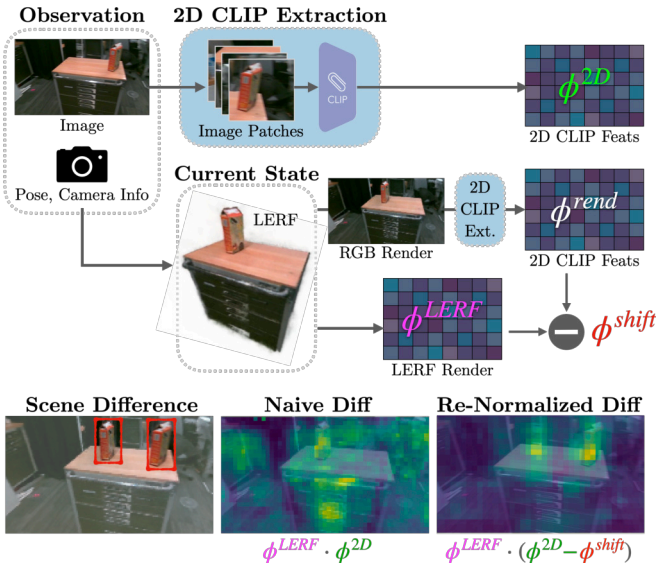


Fig. 3: *Semantic differencing*. The semantic differencing module calculates 2D feature maps from the fresh observation (top), the 3D LERF embeddings (middle), and the 2D CLIP embeddings of a NeRF-rendered image. $\phi^{rend}$ approximates the distribution shift from 2D to 3D, resulting in higher-quality semantic difference heatmaps (bottom). See Sec IV-B for details.

the scene, dynamically adding images during training as opposed to training on a fixed set. Second, we compute CLIP and DINO features in parallel with the ongoing LERF optimization, computing features in a callback and offloading completed images lazily within the train loop. This requires significant computation, about 500ms on a 4090 GPU per image, motivating the usage of cloud robotics to offload this process.

### B. Scene Change Detection

Motivated by the constantly changing environments in homes and warehouses, Lifelong LERF detects changes in the environment over time so that it can selectively update these regions. Given an image and approximate camera pose as input, the output of this module is a set of 3D bounding boxes of changed regions. We explore a semantic differencing method based on language embeddings rendered from CLIP for computing 3D changed regions. This method leverages the novel view synthesis ability of implicit neural fields for comparing the current reconstruction to new input views.

*1) 3D LERF Feature Difference:* Since the primary use case of a semantic map is for semantic queries, this approach uses the underlying language embeddings to identify scene changes. To detect semantic changes, we first render the 3D field from the same camera pose as the new image to obtain a 2D feature map $\phi^{LERF}$. Since LERF is a multi-scale representation, we pick an image scale of 0.25 for querying the underlying field with and use the same scale to generate 2D embeddings by sliding a crop over the input image and passing the crops through the CLIP image encoder to obtain $\phi^{2D}$.

Naively computing the difference $\phi^{LERF} - \phi^{2D}$ falsely activates on edges of the scene. This is because the CLIP embeddings in 3D experience a distribution shift from their respective input views. To understand this, consider a scene with a tall narrow object: input views of this object contain primarily the background so their CLIP embeddings are dominated by background features. However, the background is inconsistent between different viewing angles, meaning the average embedding in 3D will be different than each 2D view. To overcome this, we propose a method for estimating this 2D-to-3D distribution shift and apply it to obtain a *semantic re-normalization* of differences obtained. Let $\phi^{rend}$ be the 2D grid of embeddings obtained by calculating CLIP embeddings of sliding a crop window over a *rendered* RGB image from NeRF. Then, the distribution shift can be quantified as $\phi^{LERF} - \phi^{rend}$. We subtract off this shift from the 2D image embeddings, re-normalize, then compute the dot product with 3D embeddings to find the difference. Mathematically, this is

$$\texttt{renorm}(\phi^{2D} - (\phi^{LERF} - \phi^{rend})) \cdot \phi^{LERF}$$

This re-normalization is critical; visualizations of an ablation are provided in Fig.3.

*2) 3D Box Detection:* We utilize the semantic differencing module while the robot drives to estimate 3D changed regions. To do this, we binarize heatmaps with a threshold of 0.9, with values below 0.9 cosine similarity registering as different. We then run Tarjan's connected component's algorithm [77] to parse the heatmap to discrete heatmaps that do not intersect. We then filter out heatmaps with less than 30 pixels and deproject each heatmap into 3D by using the minimum of NeRF depth and RealSense depth, to account for the potential of adding or removing objects. We concatenate point clouds from 15 images at a time, then compute clusters using DBSCAN and fit oriented bounding boxes to each cluster. We constrain boxes' Z axes to align with the world axis.

### C. Updating LERF

Once changes have been identified in 3D, we must selectively update both the language and visual properties of the LERF to match the scene state. To do this, we mask input NeRF views from the previous stage by projecting the 3D bounding boxes into each training view. We then prevent rays from being sampled inside image masks, which effectively prevents the LERF from incorporating information from stale regions of images. During each time period of exploration, we only mask images from prior time periods to avoid over-masking images. NeRF leverages *proposal networks* [40] to guide the volumentric sampling process, which represents a low-frequency 3D distribution of where geometry lies in the scene. We modify the proposal sampling procedure by adding a constant probability mass of 0.02 to each ray sample, determined empirically, ensuring that samples are always taken in free space. This is important for convergence in locations where objects are added as otherwise the field would never be sampled where objects are added.


Fig. 4: **Experiment setup** *(Left)*: Test objects; (*Right*): Three types of scene changes included in evaluation. Red box denotes the changed scene region.

### D. Fog-ROS2 Integration

Wheeled mobile robots possess limited computation power designed for basic controllers and sensors. Consequently, they cannot execute Lifelong LERF on their single-board computer, while both DROID-SLAM and LERF necessitate distinct GPUs. To overcome this limitation, we incorporate Fog-ROS2 to facilitate access to on-demand cloud computation via a separate workstation. We encapsulate DROID-SLAM and LERF in ROS2 nodes and implement the publish/subscribe ROS2 interfaces; consequently, the robot can publish images to DROID-SLAM in the cloud, which consecutively publishes the maps and keyframe poses with LERF.

## V. Experiments

To evaluate Lifelong LERF we consider a local table with up to 5 objects lying on it (see Fig. 2). We use a Turtlebot 4 equipped with a RealSense D457 camera which faces toward the starboard side. SLAM and LERF optimization executes on a workstation with 2 RTX 4090s (Fig. 2). This direction is chosen to maximize camera parallax to enhance SLAM pose estimation and NeRF quality. In addition, it makes mapping inward-facing scenes easier with a differential drive base. Experiments progress in *time periods*: we first capture the entire scene by driving the robot in a fixed trajectory around the table and train the LERF for 2000 steps (2 minutes) before initiating the next time period. All consecutive queries happen without additional idle training time, only updating the LERF during robot motion.

In subsequent time periods, we randomly select one or two objects to add, remove, or replace within the scene (Fig. 5), and the robot executes a partial trajectory to detect changes and decide whether to remap or not. If it detects changes, it completes its trajectory around the table and adds new images to the LERF, otherwise it early-terminates. Between each time period, a language query for each object is sent to the LERF; both updated and static objects. A language query is considered a success if the argmax language activation in the 3D scene lies on the correct object. For objects that were removed, we consider the localization a success if the argmax language activation no longer lies on its previous location, and for swapped objects if a query for the removed object no longer activates on swapped in object. LERF is well known to struggle with answering "existence" questions [4], so to evaluate how well the language embeddings are replaced we ensure removed objects' language queries no longer remain in the same location, and maintain the same activation as the background.

The experiments divide possible changes in a scene into 4 categories:

1) *No Change*: after the initial scan, no objects are removed from the scene. This measures a scene difference detector's bias towards false positive prediction (i.e. model predicting that the scene is changed and needs to be remapped but instead is unchanged).
2) *Removal*: after the initial scan, one or two objects are removed from the scene.
3) *Addition*: after the initial scan, one or two objects are added to the scene.
4) *Swap*: one or two objects are placed in the scene after the initial scan.

For every experiment with scene changes, we execute 2 difficulty tiers per category and 2 trials per tier. Tier 1 moves 1 object between each time period, and Tier 2 moves 2 objects between each time period. All experiments consist of 3 time periods, except the no change experiment which has 2.

During experiments for add, remove, and swap we measure 1) the percent of changed objects detected by Lifelong LERF, defined by outputting bounding boxes fully containing them, 2) the number of pixels masked in the original image, and 3) language query accuracy for each object on the table after each time period. We aim for recall to be as close to 100% while minimizing the number of masked pixels, preserving the original input images wherever possible. We separately calculate language query accuracy for moved and static objects, to separately evaluate the robustness of the LERF updating method for adapting language embeddings without affecting existing regions of the scene. Finally, to test the false positive rate of the system, we execute the same trajectory twice and measure whether the system falsely detects a change in the scene.

### A. Baseline: Depth-Camera Image Differencing

We compare against depth-based image differencing where the robot renders depth from the stored LERF at the same pose as the incoming image and calculates the pixel-wise absolute difference between the two depth images, then filter differences which are larger than 30cm or smaller than 10cm, and rejects depth values outside the range 10cm to 150cm. The resulting binary heatmap is passed through the same pipeline described in Sec IV-B. Note that both the baseline and lifelong LERF models have the same language query time and change detection time, since they share the same LERF architecture and make the same number of 2D renders.

## VI. Results

We average the detection percent and the localization accuracy across 4 different scenes for each category of scene change and summarize the results in Tables I, II. Empirically, the depth-differencing baseline tends to mistakenly identify most objects in the scene as changed because of subtle camera pose drift over time. Any pose misalignment

| Change Detection Method | Add | | Remove | | Swap | |
|---|---|---|---|---|---|---|
| | Baseline | Lifelong LERF | Baseline | Lifelong LERF | Baseline | Lifelong LERF |
| Language Query Accuracy (Moved Objects) (%) | 75 | **92** | 89 | **90** | 67 | **92** |
| Language Query Accuracy (Static Objects)(%) | 83 | **90** | 97 | 97 | 63 | **75** |
| Pixel Mask Ratio (%) | **13** | 21 | 20 | **17** | **17** | 24 |
| Change Detection Recall (%) | 92 | 92 | 83 | **92** | 67 | **92** |

TABLE I: **Results**: We evaluate the two scene change detection methods across 8 scenes. Lifelong LERF has on average an 91% language query on accuracy objects moved throughout the trial, highlighting its ability to adapt both geometry and language as the scene changes. Both depth and semantic differencing can detect scene changes with high recall with similar pixel mask ratios, but semantic differencing is more robust against false positives (see Tab II).

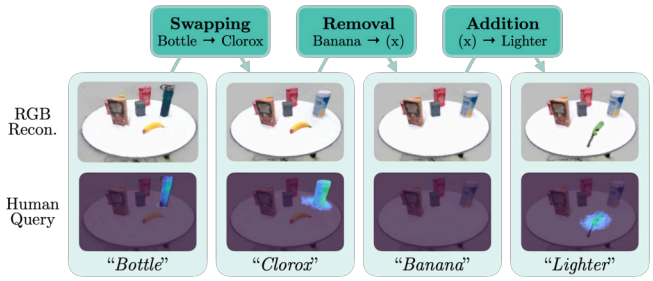


Fig. 5: *Sequential scene update.* Scene reconstructions are shown in the middle, and human queries are shown at the bottom. As the scene updates, the heatmap for human queries updates accordingly.

| Change Detection Method | No Change | |
|---|---|---|
| | Baseline | Lifelong LERF |
| Decision Accuracy | 0% | **80%** |
| Pixel Mask Ratio | 8% | **1.2%** |

TABLE II: **False Positive Evaluation**: Depth over-predicts due to SLAM pose drift and NeRF frame misalignment, leading to high false positives. Lifelong LERF is more pose-error tolerant.

produces false positives in the depth difference and results in a heatmap that over-segments objects.

### A. No Changes

Because depth is sensitive to pose misalignment, the depth difference method falsely detects changes every time. Lifelong LERF successfully predicts that there are no changes within the first 15 images 80% of the time. This saves substantial time since the robot does not need to fully remap the environment. Lifelong LERF is more robust to pose misalignment because this language difference is computed at a lower spatial frequency due to CLIP's [5] receptive field.

### B. Adding and Removing

When adding objects, the depth-differencing baseline averages 79% language query accuracy and a 92% change detection recall rate. Lifelong LERF averages 91% language query accuracy and detects changes in the environment correctly 92% of the time. The depth-based baseline tends to under-select added objects, leading to insufficient updates to the LERF to update the 3D semantic features, resulting in slightly worse language performance. In the object removal setting, the depth-based baseline achieves comparable language query performance with Lifelong LERF. Prior work [7] has also demonstrated that it is comparatively easier to remove information from NeRF than to add new information.

### C. Swapping Objects

The depth baseline suffers significantly because swapping an object largely preserves scene geometry, while the semantic differencing method is able to capture the differences in the scene since the language embeddings between new and old objects differ significantly. This is reflected by the fact that the baseline localizes moved objects with 67% accuracy, while semantic differencing localizes a 92%.

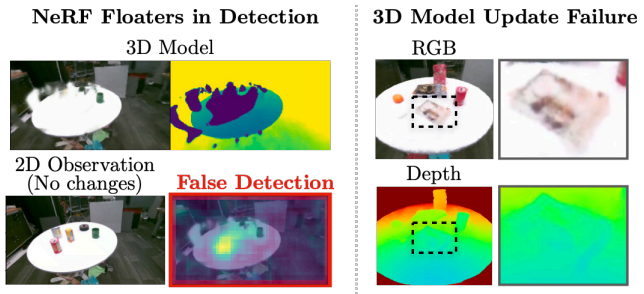


Fig. 6: **Lifelong LERF failure modes:** *(Left):* NeRF's spurious density triggers false difference detection and alters CLIP feature differences through incorrect activations. *(Right):* If semantic changes are not correctly detected, dataset inconsistency will form fuzzy density filled with holes and fail to fully add or remove the object.

## VII. Limitations

Some of the failure modes are presented in Fig. 6. Lifelong LERF operates under the assumption of a known scene and uses pre-defined trajectories for mapping. It is not practical for highly dynamic environments and large-scale scenes, which may benefit from autonomous exploration and multi-camera setups. The algorithm also shows sensitivity to object size in semantic change detection and has a tendency to over-segment scenes. More sophisticated approach to difference detection which combines segmentation, semantics, geometry, and perhaps a learned difference detection module may be interesting future work.

## VIII. Conclusion

In this study, we introduce Lifelong LERF, a system that builds and updates a LERF to dynamically adapt to semantic changes in a scene. Using the dense semantics of LERF, we propose a method for detecting scene changes that is more robust to pose errors or reconstruction inaccuracy compared to a depth camera differencing baseline. Results suggest high accuracy in change detection with a minimal rate of false positives in local tabletop settings, and the ability to rapidly query the updated LERF with natural language to inventory objects. We offload computation to a local server using Fog-ROS2 to enable deployment on an inexpensive Turtlebot robot. Future work will study scaling this method to larger scale room environments, where more complex mapping is necessary.

## REFERENCES


[1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in neural information processing systems*, 2021.

[4] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *International Conference on Computer Vision (ICCV)*, 2023.

[5] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[7] J. Kerr *et al.*, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*, 2022.

[8] K. Chen *et al.*, "FogROS2-SGC: A ROS2 Cloud Robotics Platform for Secure Global Connectivity," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 2035–2042.

[9] A. Rashid *et al.*, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023.

[10] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*, IEEE, 2018, pp. 32–41.

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[12] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9031–9040.

[13] J. Hou, A. Dai, and M. Nießner, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4421–4430.

[14] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald, "3d instance segmentation via multi-task metric learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9256–9266.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[16] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[17] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 1689–1696.

[18] I. Armeni *et al.*, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[19] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7515–7525.

[20] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[21] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *International Conference on Learning Representations*, 2022.

[22] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision*, Springer, 2022, pp. 540–557.

[23] F. Liang *et al.*, "Open-vocabulary semantic segmentation with mask-adapted clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7061–7070.

[24] A. Kirillov *et al.*, "Segment anything," *arXiv:2304.02643*, 2023.

[25] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello, "ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models," *arXiv preprint arXiv: 2303.04803*, 2023.

[26] K. Jatavallabhula *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *rss*, 2023.

[27] J.-M. Park, J.-H. Jang, S.-M. Yoo, S.-K. Lee, U.-H. Kim, and J.-H. Kim, "Changesim: Towards end-to-end online scene change detection in industrial indoor environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 8578–8585.

[28] L. Ru, B. Du, and C. Wu, "Multi-temporal scene classification and scene change detection with correlation based fusion," *IEEE Transactions on Image Processing*, vol. 30, pp. 1382–1394, 2020.

[29] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 4063–4067.

[30] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[31] S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, and L. Schmid, "3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 8179–8186.

[32] L. Schmid *et al.*, "Panoptic multi-tsdfs: A flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 8018–8024.

[33] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense rgb-d maps," in *2013 European Conference on Mobile Robots*, IEEE, 2013, pp. 178–185.

[34] M. Fehr *et al.*, "Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *2017 IEEE International Conference on Robotics and automation (ICRA)*, IEEE, 2017, pp. 5237–5244.

[35] E. Langer, T. Patten, and M. Vincze, "Robust and efficient object change detection by combining global semantic information and local geometric verification," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 8453–8460.

[36] M. T. Lázaro, R. Capobianco, and G. Grisetti, "Efficient long-term mapping in dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 153–160.

[37] T. Krajník, J. P. Fentanes, M. Hanheide, and T. Duckett, "Persistent localization and life-long mapping in changing environments using the frequency map enhancement," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4558–4563.

[38] M. Adamkiewicz *et al.*, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[39] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[40] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[41] L. Ma *et al.*, "Deblur-nerf: Neural radiance fields from blurry images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 861–12 870.

[42] X. Huang, Q. Zhang, Y. Feng, H. Li, X. Wang, and Q. Wang, "Hdr-nerf: High dynamic range neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 398–18 408.

[43] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "Robustnerf: Ignoring distractors with robust losses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 626–20 636.

[44] J. Philip and V. Deschaintre, "Radiance field gradient scaling for unbiased near-camera training," *arXiv preprint arXiv:2305.02756*, 2023.

[45] M. Tancik *et al.*, "Nerfstudio: A modular framework for neural radiance field development," *SIGGRAPH*, 2023.

[46] P. Wang *et al.*, "F2-nerf: Fast neural radiance field training with free camera trajectories," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4150–4159.

[47] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-nerf: Anti-aliased grid-based neural radiance fields," *arXiv preprint arXiv:2304.06706*, 2023.

[48] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[49] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 333–350.

[50] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.

[51] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," *arXiv preprint arXiv:2112.05131*, 2021.

[52] K. Park *et al.*, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *ACM Trans. Graph.*, vol. 40, no. 6, Dec. 2021.

[53] Z. Li, Q. Wang, F. Cole, R. Tucker, and N. Snavely, "Dynibar: Neural dynamic image-based rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4273–4284.

[54] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[55] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "Imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[56] Z. Zhu *et al.*, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[57] D. Lisus, C. Holmes, and S. Waslander, "Towards open world nerf-based slam," in *2023 20th Conference on Robots and Vision (CRV)*, IEEE, 2023, pp. 37–44.

[58] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[59] C.-M. Chung *et al.*, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9400–9406.

[60] R. Po, Z. Dong, A. W. Bergman, and G. Wetzstein, "Instant continual learning of neural radiance fields," 2023.

[61] Z. Yan, Y. Tian, X. Shi, P. Guo, P. Wang, and H. Zha, "Continual neural mapping: Learning an implicit scene representation from sequential observations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 782–15 792.

[62] Z. Cai and M. Mueller, *Clnerf: Continual learning meets nerf*, 2023. arXiv: 2308.14816 [cs.CV].

[63] J. Fu, Y. Du, K. Singh, J. B. Tenenbaum, and J. J. Leonard, "Robust change detection based on neural descriptor fields," 2022.

[64] Y.-L. Liu *et al.*, "Robust dynamic radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13–23.

[65] M. A. Karaoglu *et al.*, "Dynamon: Motion-aware fast and robust camera localization for dynamic nerf," *arXiv preprint arXiv:2309.08927*, 2023.

[66] Y. Yin, Z. Fu, F. Yang, and G. Lin, "Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields," *arXiv preprint arXiv:2305.10503*, 2023.

[67] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[68] J. Ichnowski *et al.*, "Fog robotics algorithms for distributed motion planning using lambda serverless computing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4232–4238.

[69] N. Tian *et al.*, "A cloud robot system using the dexterity network and Berkeley robotics and automation as a service (BRASS)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1615–1622.

[70] P. Li, B. DeRose, J. Mahler, J. A. Ojea, A. K. Tanwani, and K. Goldberg, "Dex-Net as a service (DNaaS): A cloud-based robust robot grasp planning system," in *IEEE Conference on Automation Science and Engineering (CASE)*, 2018, pp. 1420–1427.

[71] *AWS IoT Greengrass*, `https : / / aws . amazon . com / greengrass/`, Accessed: 2021-02-15.

[72] G. Mohanarajah, D. Hunziker, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 481–493, 2014.

[73] K. E. Chen *et al.*, "FogROS: An adaptive framework for automating fog robotics deployment," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2021, pp. 2035–2042.

[74] J. Ichnowski *et al.*, "FogROS2: An adaptive and extensible platform for cloud and fog robotics using ros 2," *arXiv preprint arXiv:2205.09778*, 2022.

[75] K. Chen, J. Yuan, N. Jha, J. Ichnowski, J. Kubiatowicz, and K. Goldberg, "FogROS G: Enabling secure, connected and mobile fog robotics with global addressability," *arXiv preprint arXiv:2210.11691*, 2022.

[76] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[77] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.