2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Language-Embedded Gaussian Splats (LEGS): Incrementally Building Room-Scale Representations with a Mobile Robot

Justin Yu*[1], Kush Hari*[1], Kishore Srinivas*[1] Karim El-Refai[1], Adam Rashid[1], Chung Min Kim[1], Justin Kerr[1], Richard Cheng[2], Muhammad Zubair Irshad[2], Ashwin Balakrishna[2], Thomas Kollar[2], Ken Goldberg[1]

***Abstract*— Building semantic 3D maps is valuable for searching for objects of interest in offices, warehouses, stores, and homes. We present a mapping system that incrementally builds a Language-Embedded Gaussian Splat (LEGS): a detailed 3D scene representation that encodes both appearance and semantics in a unified representation. LEGS is trained online as a robot traverses its environment to enable localization of open-vocabulary object queries. We evaluate LEGS on 4 room-scale scenes where we query for objects in the scene to assess how LEGS can capture semantic meaning. We compare LEGS to LERF [1] and find that while both systems have comparable object query success rates, LEGS trains over 3.5x faster than LERF. Results suggest that a multi-camera setup and incremental bundle adjustment can boost visual reconstruction quality in constrained robot trajectories, and suggest LEGS can localize open-vocabulary and long-tail object queries with up to 66% accuracy. See project website at: berkeleyautomation.github.io/LEGS**

I. Introduction

Consider open vocabulary robot requests such as *"Where are gluten-free crackers?"* or *"Get a stain remover spray"*, the robots must parse such queries, localize relevant objects, and navigate to them. A large body of recent work uses large vision-language models by distilling their outputs into 3D representations like point clouds or NeRFs [2]. These semantic representations have been applied to both manipulation [3], [4], [5] and large-scale scene understanding [6], [7], [8], showing promise of using large models zero-shot for open-vocabulary task specification.

One key challenge for scaling these methods to large environments is the underlying 3D representation, which should be flexible to a variety of scales, able to update with new observations, and fast. Although NeRFs are commonly used as the 3D representation for distilling 2D semantic features [9], [10], [1], scaling NeRFs to large scenes can be cumbersome because they typically rely on a fixed spatial resolution [11], [12], [13], are difficult to modify, and slower to render. A popular alternative is pointclouds [14], [7], [8], [6], which work seamlessly with many SLAM algorithms. However, a given point is assigned a single color and semantic feature by fusing CLIP in the pointcloud with a contrastively supervised field, whereas a multi-scale model of the world can simultaneously reason about objects and their parts, similar to how LERF-TOGO [5] leverages multi-scale semantics in LERF [1].

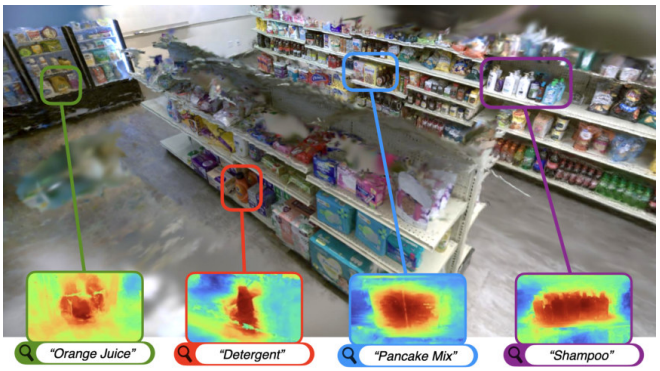

Fig. 1: **Language-Embedded Gaussian Splat in TRI Grocery Store Testbed [15].** LEGS relies entirely on pretrained VLMs and does not require any inventory data or finetuning.

3D Gaussian Splatting (3DGS) [16] models the 3D scene using a large set of 3D Gaussians. Recent works [17], [18] successfully assign semantic features to every Gaussian in the scene. However, existing techniques combining semantic features and 3D Gaussian Splatting (3DGS) scene reconstruction require offline computation of keyframe transforms and 3D Gaussian initialization points.

In this paper, we focus on linking language understanding to Gaussian Splats in large-scale scenes, while incrementally training on a stream of RGBD images of the scene from a mobile robot. This incremental training method offers substantial benefits, notably enabling the robot to autonomously determine its position within the environment and subsequently use the map data for enhanced operational efficiency.

LEGS combines geometry and appearance information from 3DGS with semantic knowledge from CLIP by grounding language embeddings into the 3DGS similar to the method described in [17]. LEGS incrementally registers images and simultaneously optimizes both 3D Gaussians and dense language fields. This allows robots to build maps that contain rich representations of their surroundings that can be queried with natural language.

This paper makes 3 contributions:

- An online multi-camera 3DGS reconstruction system for large-scale scenes. The system takes as input three video streams from a mobile robot, and incrementally builds the 3D scene.
- Language-Embedded Gaussian Splatting (LEGS), a hybrid 3D semantic representation that uses explicit 3D Gaussians for geometry and implicit scale-conditioned

* Equal contribution
[1]The AUTOLab at UC Berkeley (automation.berkeley.edu).
[2]Toyota Research Institute, Los Altos, CA.

hashgrid [19] for the semantics.
- Results from physical experiments suggesting LEGS can produce high quality Gaussian Splats in room-scale scenes with training time 3.5x faster than a LERF baseline [1].

II. Related work

A. Mobile Robot Mapping

Early robotic scene mapping research focused on the development of the core competencies in the metric [20], [21], [22] and topological [23], [24] knowledge spaces, extensively centered around the question of map and knowledge representation. For successful task execution, data-rich 3D scene representation and self-localization are critical, enabled by Simultaneous Localization and Mapping (SLAM) algorithms [25], [26], [27]. 3D spatial maps have traditionally been represented by voxel grids, points or surfels, and more recently, neural radiance fields [28], [29]. Each of these approaches come with their own limitations. The accuracy and expressiveness of occupancy and voxel grids are resolution-bounded due to quantization. Points and surfels are discontinuous when rendered, making it challenging to supervise features in a continuous manner. Recent SLAM methods adopting a neural radiance field representation such as NICE-SLAM [30] and NeRF-SLAM [31] are constrained by their implicit representation making it difficult to update geometry over time.

B. Semantic Scene Mapping for Robotics

Semantic grounding, particularly in 3D representations, is a longstanding problem [32] to integrate semantic knowledge of objects and the surrounding environment into a mapped scene. The first definition of semantic mapping for robotics is provided by Nüchter, *et al.* as a spatial map, 2D or 3D, augmented by information about entities, i.e., objects, functionalities, or events located in space [33]. An early work proposes concurrent object identification and localization using a supervised hierarchical neural network classifier on image color histogram feature vectors [34]. However, because these approaches rely on supervised datasets [35], [36], they work only on a closed set of vocabularies and do not generalize to open-ended semantic queries.

More recent works have focused on using large vision-language models to support open-vocabulary queries. This includes both 2D [37], [38], [39] and 3D, such as VL-Maps [8] and CLIP-fields [7], which assigns a CLIP feature to every point in the 3D scene. This can be used for setting navigation goals with natural language queries. OpenScene [14] ensembles open-vocabulary feature encoders and 3D point networks to form a per-point feature-vector allowing natural language querying on pointclouds. ConceptFusion [40] develops 3D open-set multimodal mapping by projecting CLIP [41] and pixel aligned features into 3D points, and additionally fuse other modalities such as audio into the scene representation. ConceptGraphs [42] model spatial relationships as well as the semantic objects in the scene to reason over spatial and semantic concepts.

Semantic fields have been applied not only to scene-level understanding for mobile robots but also to manipulation. In these settings, NeRFs [2] have been a popular 3D representation, following from Distilled Feature Fields [9], Neural Feature Fusion Fields [10], and Language Embedded Radiance Fields (LERFs) [1]. These works learn an semantic field in addition to the color field. LERF supports a scale-conditioned feature field, which takes in an extra scalar as input to facilitate feature encodings at multiple scene scales. For manipulation, the feature fields have been shown to facilitate learning from few-shot demonstrations [4], policy learning [3], zero-shot trajectory generation [43], and task-oriented grasping [5].

LERF-TOGO's [5] zero-shot task-oriented grasping performance is fully based on LERF, as LERF's multi-scale semantics allow for both object- and part-level understanding. This property is also valuable in scene-level settings, where a human may specify a collection of objects, e.g., utensils. LEGS maintains this multi-scale understanding, while speeding up training and querying time, by using Gaussian Splats [16] which have a significantly faster render time.

C. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) originated in 2023 [16] to model a scene as an explicit collection of 3D Gaussians. Each Gaussian is described by its position vector μ, covariance matrix Σ, and an opacity parameter α, creating a representation that is both succinct and adaptable for static environments. The choice of 3D Gaussians over traditional point clouds is strategic; their inherent differentiability and the ease with which they can be rasterized into 2D splats enable accelerated α-blending during rendering. By avoiding volumetric ray casting employed by Neural Radiance Fields (NeRFs), Gaussian Splatting has a substantial speed advantage and can support real-time rendering capabilities.

Soon after its release, 3DGS has been applied to mapping [44], semantic mapping [45], navigation [46], and semantic fields [17], [18]. 3DGS's fast rendering time speeds up optimization, making it suitable for integrating visual SLAM and natural language queries for 3D semantic fields. 3DGS has also been demonstrated in both indoor datasets [47], [48], [49], and outdoor driving scenes with multiple cameras where all sensor data is collected before 3DGS training [50].

D. Concurrent Research

Other work in 3D Gaussian Splatting has focused on embedding language features, and separately, training online.

Learning semantic features for Gaussians have taken either one of two approaches: calculating it on-the-fly by querying a network or maintaining multi-dimensional features for each Gaussian. FMGS [17] uses multi-resolution hash encodings [19] optimized with a render-time loss to combine CLIP features with a map of 3D Gaussians. LEGS similarly utilizes a hash encoding for its feature field, however it includes scale-conditioning as opposed to averaging CLIP across scales, retaining finer-grained language understanding. On the other hand, LangSplat [18] embeds language in 3DGS

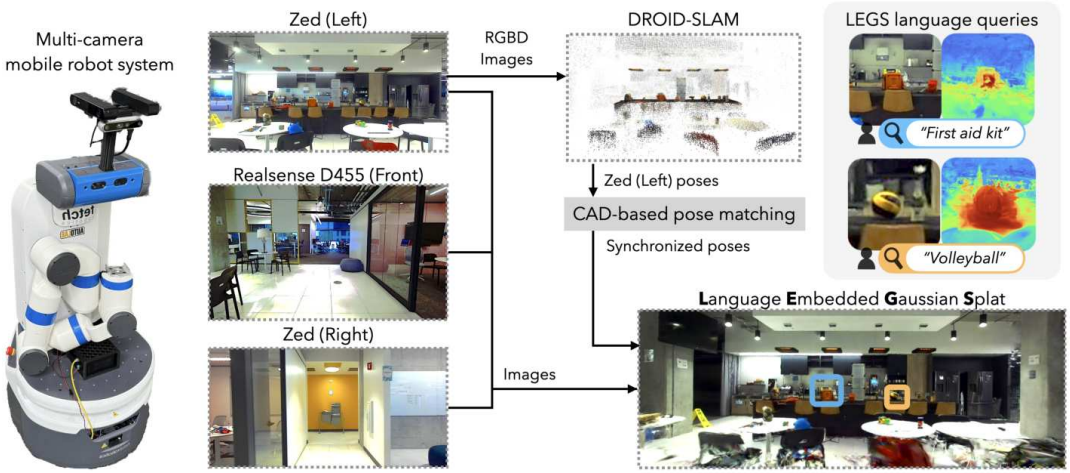

Fig. 2: **LEGS System Integration** For LEGS, we use a Fetch robot with a custom multicamera configuration where a Realsense D455 is facing forward while 2 Zed cameras face the left and right sides respectively. The left Zed image stream is inputted into DROID-SLAM to compute pose estimates for the left camera, and the corresponding extrinsics are used to compute the pose estimates for the other Zed camera and D455. These image-poses are then used for concurrent Gaussian splat and CLIP training online. From there, the Gaussian splat can be queried for an object (ex. "First Aid Kit"), and the corresponding relevancy field will be computed to localize the desired object.

by training a scene-specific autoencoder to map between CLIP embeddings and a lower-dimensional latent feature associated with each 3D gaussian. Like traditional radiance field methods, LangSplat assumes poses are corresponded with all scene images prior to 3DGS training as it requires training a VAE over all images of a scene before starting its 3D optimization. However, for robotic systems, it is often desirable to develop 3D semantic understanding online as the robot explores new and previously unseen large-scale environments.

SplaTAM [44] optimizes both camera pose and the 3D Gaussian map simultaneously for single-camera setups. However, having multiple cameras and viewpoints can enhance efficient environment data collection. Additionally, SplaTAM lacks semantic features, which is important for identifying and interacting with objects in a 3D scene.

To our knowledge, LEGS is the first system that integrates the advantages of both online 3DGS training and language-aligned feature supervision into Gaussian splats for large-scale scene understanding.

III. Problem Statement

We consider a large indoor environment, specifically defined as a room encompassing at least 750 sq ft. The objective is to 3D reconstruct the a-priori unknown and unstructured environment and localize objects prompted by open-vocabulary and long-tail natural-language queries.

We make the following assumptions:

1) The environment and all objects within it are static.
2) Queried objects are seen at least from one of the cameras.
3) A mobile robot with 3 orthogonal cameras. For the grocery store environment, the TTT robot with a single pair of stereo cameras is used [15].

For each trial, the system is prompted by a natural-language query, and outputs the heatmap and localized 3D coordinate of the most semantically relevant location in the scene. The trial is deemed successful if this point falls within a manually annotated bounding box for that object. The objective is to efficiently build a 3D representation that maximizes this success rate for large-scale scenes.

IV. Methods

We use a Fetch mobile robot equipped with an RGB-D Realsense D455 camera and two side facing ZED 2 stereo cameras mounted with known relative poses. We build a map of the scene with a set of 3D Gaussians in an online fashion, registering new images as the robot drives around the environment. The overall pipeline for LEGS is outlined in Figure 2. There are three key components of the system:

1) **Multi-Camera Reconstruction**: To improve the effective field-of-view of the robotic system, we use multiple cameras pointing in different directions to provide more viewpoints of the environment.
2) **Incremental 3DGS Construction**: A significant challenge of large scene mapping is localization error due to its accumulative nature. To mitigate this error, we perform global Bundle Adjustment (BA) with DROID-SLAM to improve pose accuracy for all previously recorded poses in the scene. Global BA can be executed multiple times in a given traversal, and after each BA, the prior image-pose estimates are updated with the corresponding pose.
3) **Language-Embedded Gaussian Splatting**: We implement a language-aligned feature field inspired by the method from LERF [1] that samples from gaussian primitives instead of from a density field.

A. Multi-Camera Reconstruction

Online image registration enables Gaussian Splatting on a mobile base with a comprehensive sensor suite including multiple camera views (left, right, center). During testing of vanilla Gaussian Splatting on our multi-camera setup, we found that offline Structure from Motion (SfM) pipelines frequently failed to find correspondences between images from different cameras, largely due to the lack of scene overlap from offset camera views. Because we peform online

image registration with a visual SLAM algorithm for one camera and we know the corresponding extrinsic transforms to the other cameras, we can compute the corresponding pose estimate for each camera. Due to limited GPU memory onboard the Fetch robot, the image data is then streamed over network to a desktop computer for processing and training.

B. Incremental 3DGS Construction + Bundle Adjustment

Standard methods for radiance field optimization require image-pose pairs as input, and Gaussian Splatting greatly benefits from having a pointcloud as a geometric prior for initialization. Poses and pointclouds are typically provided through *offline* Structure from Motion (SfM) techniques like COLMAP [51] which require all training images to be collected ahead of time. We build off of Nerfstudio's [52] Splatfacto implementation of Gaussian Splatting and modify it to operate on a stream of images and poses, and incorporate updates from global bundle adjustment (BA) to further optimize poses by building a keyframe graph to minimize the Mahalanobis distance between the reprojected points and the corresponding revised optical flow points [53].

1) Online Optimization: For online pose estimation we use DROID-SLAM [53], a monocular SLAM method that takes in monocular, stereo, or RGB-D ordered images and outputs per-keyframe pose estimates and disparity maps. During operation, we feed DROID-SLAM input frames from one of the side-facing Zed cameras, and extrapolate the poses of other cameras using the camera mount CAD model. Registered RGBD keyframes from DROID-SLAM are incrementally added to Splatfacto's training set. We initialize new Gaussian means per-image by sampling 500 pixels from each depth image and deproject them into 3D using the corresponding metric depth measurement. We use a learned stereo depth model trained on synthetic images [54], which can predict thin features and transparent objects with high accuracy.

2) Global Bundle Adjustment: Incorporating images with pose drift from SLAM systems results in artifacts in 3DGS models like duplicated or fused objects, fuzzy geometry, and ghosting (Fig. 4). Though prior work has demonstrated that pose optimization inside a 3DGS can track camera pose [44], tracking iterations for the method with reported results takes up to a second to perform, making it difficult for online usage. Instead, to mitigate drift, we incorporate updates from global BA and update training camera poses in the 3DGS accordingly. This allows tracking of new camera frames at 30fps in tandem with continual 3DGS optimization for faster model convergence.

C. Language Embedded Gaussian Splats

In Language Embedded Radiance Fields (LERF) [1] the language field is optimized by volumetrically rendering and supervising CLIP embeddings along rays during training. In contrast, 3DGS provides direct access to explicit gaussian means, allowing us to implement a multi-scale language embedding function, $F_{lang}(\vec{x}, s) \in \mathbb{R}^D$. This function takes an input position $\vec{x}$ and physical scale s, outputting a D-dimensional language embedding. F_{lang} is implemented by passing a sampled $\vec{x}$ through a multi-resolution hash encoding [19], which produces the input z to the MLP $m_\theta(z, s)$, with the MLP evaluated last resulting in feature output $y \in \mathbb{R}^D$. These features can then be projected and rasterized into feature images using Nerfstudio's [52] tile-based rasterizer implementation, with loss gradients backpropagated through the MLP. Hash encoding employs a hash table to store feature vectors corresponding to grid cells in a multi-resolution fashion. This hash grid encoding scheme effectively reduces the number of floating-point operations and memory accesses needed during training and inference, compared to directly feeding position features into an MLP.

Fig. 3: **4 Scene Environments.**

Language-aligned features are obtained from the training images using multi-scale crops passed through the CLIP encoder, a technique shown in LERF to be crucial for semantic understanding in large scenes where object sizes may vary drastically. This is in contrast to prior work which averages CLIP embeddings as a tradeoff between speed and accuracy [17]. LEGS facilitates inference at approximately 50 Hz 1080p, and its hybrid explicit-implicit representation allows faster scene querying without volumetric rendering.

Given a natural language query, we query the language field to obtain a relevancy map similar to LERF. To localize the query in the world frame, we find the relevancy over CLIP features and take the argmax relevancy over 3D gaussian means. This method offers a significant speed increase over feature field methods distilled in NeRFs, where the volumetric representation must first render a dense pointcloud.

V. Experiments

A. Physical Experiments

We evaluate LEGS through a series of open-vocabulary object recall tasks. These tasks are designed to measure the system's competency in capturing and organizing information based on both location and semantic meaning. We evaluate LEGS on four large-scale indoor environments, two office kitchen scenes containing different objects, an office workspace, and a grocery store testbed [15] as seen in Figure 3. For the grocery store testbed, data is collected with the TTT robot [15]. The robot begins in a previously unseen environment and is manually pushed around a pre-planned path (including straight lines, loops, figure-8, etc) while continuously registering new images until it finishes

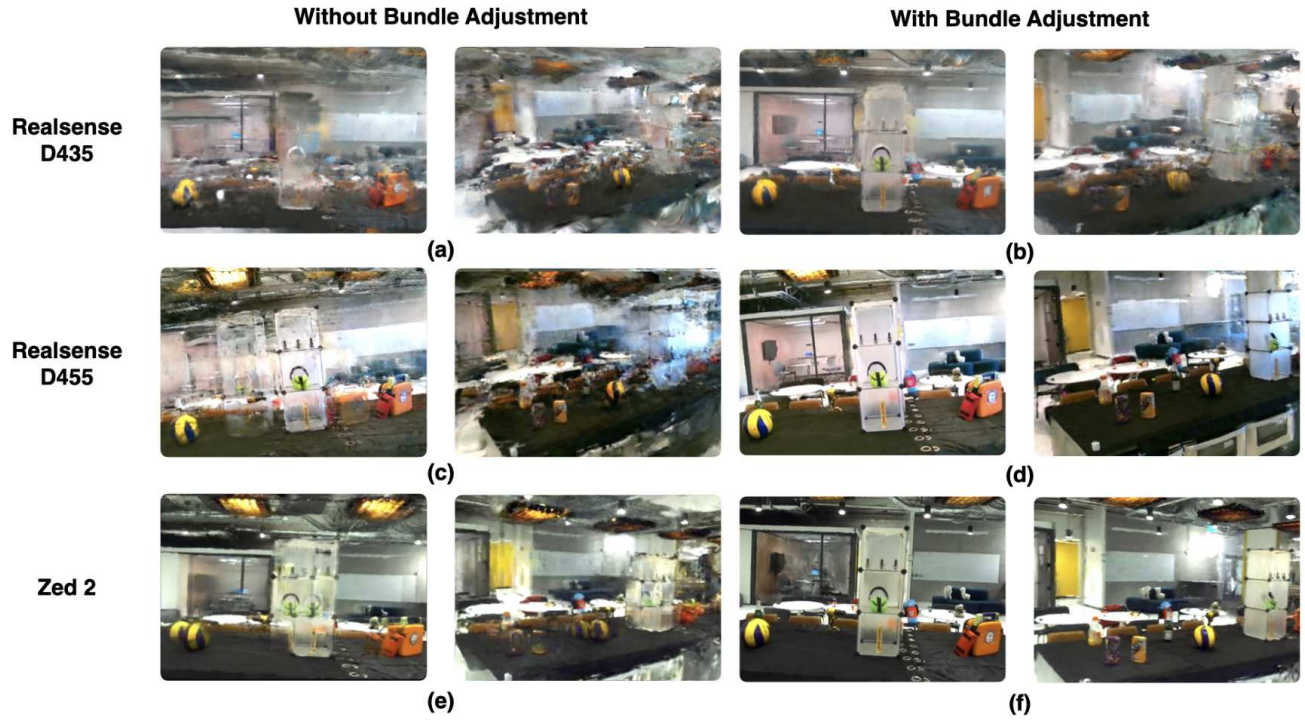

Fig. 4: **Single Camera Reconstruction Comparison Results.** We compare the quality of Gaussian splats on an Intel Realsense D435, Intel Realsense D455, and Stereolabs Zed 2 with and without bundle adjustment. For each configuration we present two views: one of the Gaussian splat facing the kitchen island head-on and another view at an angle.

the path. The robot will actuate its torso height to obtain multiple azimuthal perspectives from the same position on subsequent passes. Every 150 keyframes, we perform global bundle adjustment on all previous poses in DROID-SLAM and update accordingly in our 3D Gaussian map. Our system uses 2 NVIDIA 4090s, one for training LEGS, which takes 15 GB of memory and the other for DROID-SLAM, which can take up to 18 GB of memory.

The evaluation approach was adopted from previous 3D language mapping works [1], [55]. We randomly sample images from our training set and query Chat-GPT 4V with: "name an object in this scene in 3 words or fewer". This process is repeated until 15 unique queries are generated for each of the four scenes. We then choose a random novel view and manually annotate a 2D bounding box around each selected object of interest. Then, we query LEGS on each object and identify the highest activation energy point, and project that point in the novel 2D view. If the projected point is contained in the bounding box, we consider the query successful. Additionally, we directly baseline our approach by running LERF [1] to compare the object recall capabilities for a large-scale scene in radiance field methods.

The results in Table 1 suggest that LERF and LEGS have similar language capabilities, recalling roughly the same number of objects per scene. However, to achieve the same visual quality, LERF takes an average time of 44 minutes to train while LEGS only takes 12 minutes. Figure 5 shows examples of successful object localization queries. Localization may fail when objects are not seen well in the training views or have similar color to their background, as shown in Figure 6.

Environment	LERF	LEGS
Office Kitchen	9/15	**10/15**
Office Dining Area	**11/15**	**11/15**
Office Workspace	**10/15**	9/15
Grocery Store Testbed	**12/15**	10/15
Avg. Train Time	44min.	**12min.**

TABLE I: **Object recall success rates.** Comparison between LERF and LEGS on large scenes where both receive the same SLAM poses. LEGS receives poses incrementally, while LERF receives the final poses. Average train time refers to the time until 20 average PSNR. For LEGS we consider the time after the final image is added to the train set.

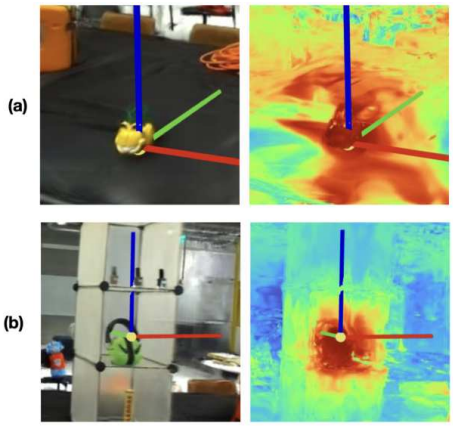

Fig. 5: **Successful query localization results.** Coordinate frames on open-vocabulary and long-tail objects (a) "garfield," (b) "hearing protection."

B. Reconstruction quality comparison

We study how camera configuration and bundle adjustment (BA) affect the quality of the LEGS Gaussian splat as

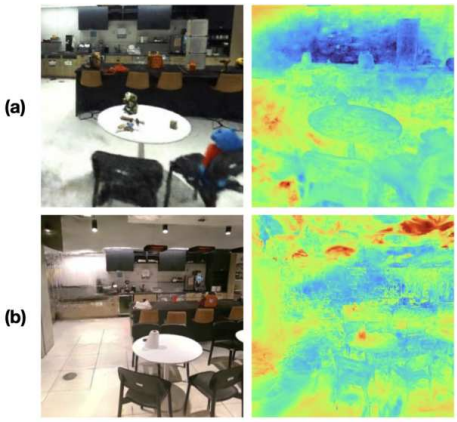

Fig. 6: **Failed query localization** on (a) "scissors," (b) "paper roll." Where object of interest is too small in the training view or lack of distinctive color features.

	PSNR	
Camera	**w/o BA**	**w/ BA**
D435	18.6	**22.7**
D455	20.0	**23.5**
Zed 2	19.2	**23.8**

TABLE II: **PSNR (Peak Signal-Noise Ratio) scores** across different cameras with and without bundle adjustment quantifying training-view reconstruction quality. Trained until 20k iterations after final image is added to the train set.

summarized in Figure 4 and Table 2. With respect to the camera configuration ablation, we evaluated different depth and stereo cameras including the Realsense D435, Realsense D455, and Zed 2. For each camera configuration, we also run a BA ablation where we either run bundle adjustment at the end of the traversal or not at all.

Global Bundle Adjustment: Table II suggests bundle adjustment improves Gaussian Splat quality for all camera configurations, removing ghostly duplicate artifacts. This is especially true for the Realsense D455 and Zed 2 cameras where the bundle adjustment configurations yielded near-photorealistic views of the scene whereas without bundle adjustment, both configurations have significantly more Gaussian floaters and/or offset objects (i.e. the left image in Figure 4 part (e) has two volleyballs). The Realsense D435 performs slightly better with bundle adjustment, but neither D435 configuration yield high quality largely due to the camera's low FOV resulting in worse localization.

We also compared a single Zed 2 camera to a multi-camera setup where the D455 Realsense is front-facing and 2 Zed cameras face the left and right side. Both gaussian splats perform well and properly render objects that were well viewed in the traversal ("raccoon toy" and "first aid kit") as seen in Figure 7. However, none of the cameras were pointing toward the ground leading to sparse views of objects near the floor. Because the multi-camera setup captures more views of the scene, it is able to construct a Gaussian splat that is better able to render these low-view objects such as the trash chutes and wet floor sign.

VI. Limitations

We assume a static environment where objects do not move during traversal. This limits the scope of this work because many applications involve dynamic scenes with moving objects. In future work, we will adapt our method to work for dynamic scenes.

Fig. 7: **Single Camera vs Multi Camera Reconstruction.** With the multi-camera setup, the effective field of view is increased, elucidating more of the scene such as the wet floor sign and trash chutes on the side faces of the table.

The motion of the Fetch mobile base can have a large effect on the LEGS reconstruction quality; the high stiction between the robot's caster wheels and the environment introduces jolts, causing camera pose inaccuracies and image blurs. In the future, we hope to correct this with a new mobile base where the trajectory is autonomously determined by a frontier-based exploration algorithm [56].

Although autonomous navigation and obstacle avoidance has been extensively studied [57], obstacles can pose a problem when it comes to the 3D Gaussian map if they are only visible in a few of the ground truth images. 3D Gaussians are initialized at the deprojected points from these few images, but there are not enough views to refine and properly train these Gaussians; the result is oddly colored floaters that obstruct some parts of the static scene.

When performing natural language queries, LEGS inherits the limitations of LERF + CLIP distillation into 3D described by similar works [1]. In our experimentation, we find that a large scale environment brings additional challenges in querying, particularly in 1) small or far-field objects in the training view, 2) similar item-background color features, such as white objects on white. Language embedded Gaussian splats can also produce false-positives when querying an object that is not in the scene due to the presence of visually or semantically similar objects, which may get incorrectly classified as the query object.

VII. Conclusion

In this work, we introduce Language-Embedded Gaussian Splats (LEGS), a system that can train Gaussian Splats online with CLIP embeddings for large-scale indoor scenes. Because of pose accumulation error that builds up in large scenes, we use incremental bundle adjustment to improve pose fidelity for Gaussian Splat training. Results suggest LEGS trains 3.5x faster than LERF with comparable object recall.

References

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

[2] 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, 2021.

[3] Y. Ze *et al.*, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in *CoRL*, PMLR, 2023, pp. 284–301.

[4] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[5] A. Rashid *et al.*, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual CoRL*, 2023.

[6] K. Jatavallabhula *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *Robotics: Science and Systems (RSS)*, 2023.

[7] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, *Clip-fields: Weakly supervised semantic fields for robotic memory*, 2023.

[8] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[9] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," *NeurIPS*, vol. 35, pp. 23 311–23 330, 2022.

[10] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations," in *2022 3DV*, IEEE, 2022.

[11] A. Meuleman *et al.*, "Progressively optimized local radiance fields for robust view synthesis," in *Proceedings of the IEEE/CVF CVPR*, 2023, pp. 16 539–16 548.

[12] 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.

[13] M. Tancik *et al.*, "Block-nerf: Scalable large scene neural view synthesis," in *CVPR*, 2022.

[14] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, *Openscene: 3d scene understanding with open vocabularies*, 2023.

[15] M. Bajracharya *et al.*, "Demonstrating mobile manipulation in the wild: A metrics-driven approach," *RSS*, 2023.

[16] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[17] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li, *Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding*, 2024.

[18] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, *Langsplat: 3d language gaussian splatting*, 2024.

[19] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, 102:1–102:15, Jul. 2022.

[20] K. O. Arras, "Feature-based robot navigation in known and unknown environments," 2003.

[21] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1985, pp. 138–145.

[22] G. Jiang, L. Yin, S. Jin, C. Tian, X. Ma, and Y. Ou, "A simultaneous localization and mapping (slam) framework for 2.5 d map building based on low-cost lidar and vision fusion," *Applied Sciences*, vol. 9, no. 10, p. 2105, 2019.

[23] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization," *IEEE Transactions on robotics and automation*, vol. 17, no. 2, pp. 125–137, 2001.

[24] A. Tapus, "Topological slam: Simultaneous localization and mapping with fingerprints of places," 2005.

[25] B. Alsadik and S. Karam, "The simultaneous localization and mapping (slam)-an overview," *Journal of Applied Science and Technology Trends*, vol. 2, no. 02, pp. 147–158, 2021.

[26] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE international symposium on safety, security, and rescue robotics*, IEEE, 2011, pp. 155–160.

[27] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 ICRA*, 2016.

[28] L. Huang, "Review on lidar-based slam techniques," in *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, IEEE, 2021, pp. 163–168.

[29] M. T. Lázaro, R. Capobianco, and G. Grisetti, "Efficient long-term mapping in dynamic environments," in *2018 IROS*, IEEE, 2018.

[30] Z. Zhu *et al.*, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF CVPR*, 2022.

[31] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," in *2023 IROS*, IEEE, 2023.

[32] L. Roldao, R. De Charette, and A. Verroust-Blondet, "3d semantic scene completion: A survey," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1978–2005, 2022.

[33] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, 2008.

[34] H. A. Kestler *et al.*, "Concurrent object identification and localization for a mobile robot," *Künstliche Intelligenz*, vol. 14, no. 4, pp. 23–29, 2000.

[35] K. Genova *et al.*, "Learning 3d semantic segmentation with only 2d image supervision," in *2021 International Conference on 3D Vision (3DV)*, IEEE, 2021, pp. 361–372.

[36] V. Vineet *et al.*, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *2015 ICRA*, IEEE, 2015.

[37] A. Brohan *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*, PMLR, 2023.

[38] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[39] B. Zitkovich *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, PMLR, 2023, pp. 2165–2183.

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

[41] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, PMLR, 2021, pp. 8748–8763.

[42] Q. Gu *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv*, 2023.

[43] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," 2023.

[44] N. Keetha *et al.*, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *CVPR*, 2023.

[45] M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, and H. Wang, *Sgs-slam: Semantic gaussian splatting for neural dense slam*, 2024.

[46] T. Chen, O. Shorinwa, W. Zeng, J. Bruno, P. Dames, and M. Schwager, *Splat-nav: Safe real-time robot navigation in gaussian splatting maps*, 2024.

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

[48] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, *Scannet++: A high-fidelity dataset of 3d indoor scenes*, 2023.

[49] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *CVPR*, 2019.

[50] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, *Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes*, 2024.

[51] S. Agarwal *et al.*, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, 2011.

[52] M. Tancik *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023*, 2023, pp. 1–12.

[53] Z. Teed and J. Deng, *Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras*, 2021.

[54] K. Shankar, M. Tjersland, J. Ma, K. Stone, and M. Bajracharya, "A learned stereo depth system for robotic manipulation in homes," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[55] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3d gaussians for open-vocabulary scene understanding," *arXiv preprint arXiv:2311.18482*, 2023.

[56] A. Topiwala, P. Inani, and A. Kathpal, "Frontier based exploration for autonomous robot," *arXiv preprint arXiv:1806.03581*, 2018.

[57] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00 022, 2017.