Persistent Object Gaussian Splat (POGS) for Tracking Human and Robot Manipulation of Irregularly Shaped Objects

Justin Yu*[1], Kush Hari*[1], Karim El-Refai*[1], Arnav Dalal[1], Justin Kerr[1], Chung Min Kim[1], Richard Cheng[2], Muhammad Zubair Irshad[2], Ken Goldberg[1]

https://berkeleyautomation.github.io/POGS

***Abstract*— Tracking and manipulating irregularly-shaped, previously unseen objects in dynamic environments is important for robotic applications in manufacturing, assembly, and logistics. Recently introduced Gaussian Splats [1] efficiently model object geometry, but lack persistent state estimation for task-oriented manipulation. We present Persistent Object Gaussian Splat (POGS), a system that embeds semantics, self-supervised visual features, and object grouping features into a compact representation that can be continuously updated to estimate the pose of scanned objects. POGS updates object states without requiring expensive rescanning or prior CAD models of objects. After an initial multi-view scene capture and training phase, POGS uses a single stereo camera to integrate depth estimates along with self-supervised vision encoder features for object pose estimation. POGS supports grasping, reorientation, and natural language-driven manipulation by refining object pose estimates, facilitating sequential object reset operations with human-induced object perturbations and tool servoing, where robots recover tool pose despite tool perturbations of up to 30°. POGS achieves up to 12 consecutive successful object resets and recovers from 80% of in-grasp tool perturbations.**

I. Introduction

In environments like factories, workshops, or homes, robots must not only successfully identify and manipulate objects but also adapt to changes in object pose over time. The challenge is greater when dealing with irregularly shaped objects for which obtaining an accurate Computer-Aided Design (CAD) model is impractical. While any physical object can in principle be CAD-modeled, this process is often labor-intensive and may require reaching out to manufacturers or purchasing specialized scanning equipment. Approaches that rely on predefined CAD models struggle in scenarios where such models are unavailable, limiting adaptability to previously unseen objects [2–4]. Traditional and deep RGBD or point cloud object tracking methods are attractive as components of state estimators for robotic manipulation because they do not require predefined meshes or CAD models [5, 6]. However, many of these approaches fail to effectively integrate geometric information across multiple object viewpoints or timesteps, and do not address the estimation or reconstruction of occluded object regions based on prior information. As a result, they struggle to maintain a persistent and holistic object representation over time.

Fig. 1: **Autonomous Object Manipulation and Tracking with POGS Unified Representation** (Top) A robot autonomously performs a pick and place primitive to move the shoe onto a shoebox given input natural language pick query "shoe" and place query "shoebox". (Bottom) A POGS unified representation enables language querying, grasp sampling, and continuous tracking of irregular objects as they move.

Implicit 3D representations like NeRFs [7] offer high-quality scene reconstructions but are ill-suited as a representation for dynamic scenes where objects may be moved and re-oriented. Recently, Gaussian Splatting [1] was introduced to create high quality 3D reconstructions by explicitly modeling scenes as a set of 3D gaussians that can be partitioned to allow rigid transforms on object-level clusters.

To enable online state estimation, tracking, and manipulation of unseen objects in dynamic environments, we present Persistent Object Gaussian Splat (POGS), an editable object-centric feature field representation with embedded language features, self-supervised visual features, and object-level grouping features to support robot manipulation. POGS uses 3D Gaussian Splatting (3DGS) to model full 3D geometry of irregular objects, allowing for continuous updates as the scene evolves.

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

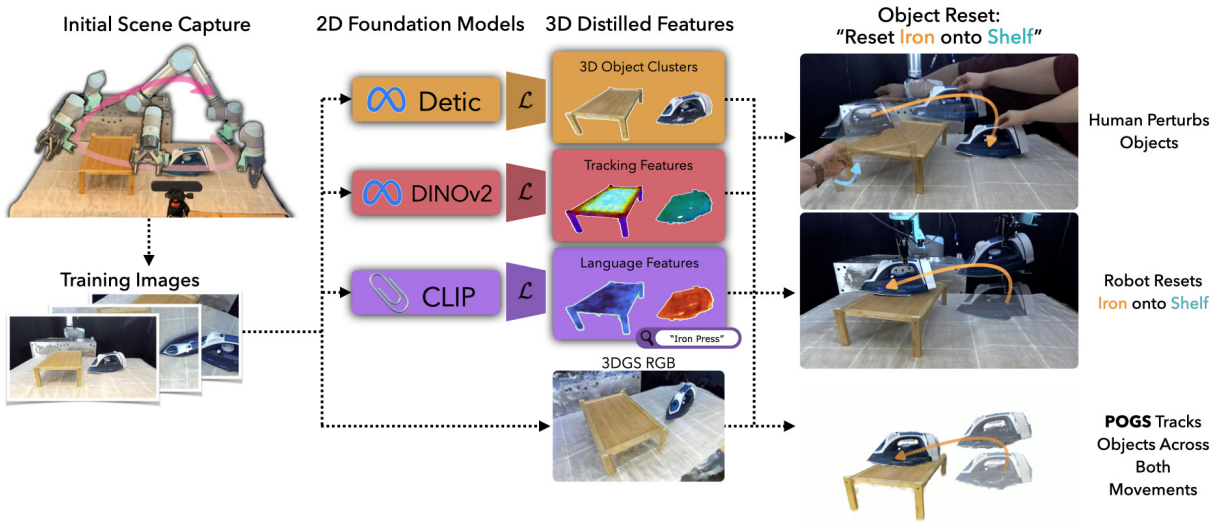

Fig. 2: **POGS Pipeline** After capturing multiple images of a scene using a robot wrist-mounted ZED mini, POGS segments objects using Detic, extracts DINO features, and embeds language through CLIP. Training images are used to optimize a 3DGS, and features extracted from 2D foundation models are distilled into feature fields, producing our POGS unified representation. During robot object reset and tool servoing, the POGS is updated based on depth geometry and DINO tracking features.

By embedding features from encoders and detectors pre-trained on internet-scale datasets such as CLIP [8], DINO [9], and Detic [10], POGS can respond to open-vocabulary natural language queries and also identify, track, and manipulate objects even without any predefined models. As such objects are moved by humans or robots, POGS can update their state online, allowing for flexible, multi-step tasks that require continuous interaction with dynamic objects, eliminating the need to re-scan the environment. This paper makes the following contributions:

- Persistent Object Gaussian Splat (POGS), a novel feature field representation for tracking and manipulating previously unseen irregularly shaped objects.
- A robot system for creating and using POGS to perform object-reset and tool servoing tasks.
- Physical robot experiments on object reset with an average pose error of 2.92 cm.
- Physical robot experiments for tool servoing where targets are moved up to 30° and the tool can recover from human perturbations 80% of the time.

II. Related Work

A. Feature Fields for Robotics

Recent advances in foundation vision models such as CLIP [8] and DINO [9] have enabled many methods that perform robot manipulation from visual features. Works [11–17] have used CLIP with point-based fusion methods to build open-vocabulary 3D representations. However, visual occlusion and multi-view pose misalignment can hinder consistent semantic fusion across the 3D scene.

To address this, more recent approaches such as DFF [18] and LERF [19] have proposed distilling learned features into neural radiance fields (NeRFs) [7] by aggregating information across multiple views and scales. F3RM [20] and LERF-TOGO [21] extend these works respectively for robot manipulation. However, NeRF-based representations are limited by NeRF's training speed and implicit spatial representation, making it impossible to update when objects move without further scene-scale optimization. Works like Dex-NeRF [22] and Evo-nerf [23] attempt to address this by partially re-scanning scenes to account for object movement; however, this process remains computationally intensive, limiting its suitability for online updates.

An alternative to NeRF is Gaussian Splatting [1], which models scenes using explicit 3D Gaussian primitives, enabling faster training and rendering while maintaining high fidelity in 3D reconstructions. Recent works [24–27] have shown that Gaussian Splatting can also integrate semantic and grouping features. GaussianGrasper [28] extends these approaches to robotic manipulation by updating the scene representation after objects are moved, using the robot end-effector pick-and-place transform followed by a few views to fine-tune the Gaussian Splat. In this work, we develop a method capable of updating the scene where a human can also move the objects repeatedly without any partial re-scans of the scene.

B. Object Tracking for Manipulation

Object pose estimation networks [29–32] are able to track the 6DOF pose of an object of interest, but typically not multiple objects at once in a scene without scaling compute requirements. Using Gaussian Splatting, several works [33–35] collect image data from one or multiple views over time for rendering dynamic scenes. However, these works focus on offline processing and pose interpolation rather than tracking and estimating object states online for manipulation tasks. Keypoint-based approaches [36–39] model multiple objects in a scene as a set of keypoints. However, these methods are prone to tracking errors when objects rotate and keypoints become occluded. Some approaches [40, 41] use multi-camera setups to help mitigate these issues. Our approach aims to achieve robust online object tracking and scene updating with

a single stereo camera.

C. Editable 3D Feature Fields

Concurrently, other works develop editable 3D feature fields. GaussianGrasper [28] and Splat-Mover [42], allow Gaussian splat updates based on known robot end-effector movements. However, they assume robot-only object interactions whereas POGS can also track human object interactions. Object-Centric Gaussian Splats [43] and GraspSplats [44] improve tracking but rely on static backgrounds or multi-camera systems. Robot See Robot Do [45] tracks part-level objects using monocular video, though only in an offline processing setting for zero-shot motion planning robot imitation from human demonstration. We extend this work to support online rigid multi-object tracking along with the aforementioned semantic and object-centric feature fields to create a unified 3D scene representation for zero-shot robotic manipulation.

III. Problem Statement

We consider a tabletop setting with irregular objects, defined as objects for which we do not have a detailed geometric (CAD) model. Given a single stereo camera, the objective is to track the 6D pose of each object over time and update the 3D scene models. We make the following assumptions:

1) Each object is rigid, simplifying the tracking to estimate rigid transforms from RGBD frames.
2) There exists an initial scanning phase in which all objects are static. However, at the start of tracking, objects can be in different poses within 90°and 25 cm of their initial scan pose.
3) All object surfaces are represented in POGS training views–with the exception of object surfaces in contact with the tabletop.
4) The scene is well-lit with approximately uniform lighting and minimal shadowing.
5) During object tracking, objects are not placed in configurations where they fully occlude each other.
6) Object surfaces exhibit low specularity for more robust geometry reconstruction and visual feature extraction.

We evaluate POGS with 2 types of robot experiments: object-reset and tool servoing.

The goal of the object-reset experiment is to use natural language to query for an object to grasp and another query for where the grasped object will be placed. After each object reset, a human will randomly reconfigure both objects to different poses and the process is repeated until failure. We evaluate this experiment by recording the maximum number of sequential object resets before failure, the object grasp rate, the object place rate, and the object translation error in placement.

In tool servoing, the objective is for the robot to continuously align a grasped tool with a target, even as a human operator moves the target and alters the tool's orientation within the robot's grip. We evaluate this experiment by recording the success rate and average time taken to recover from in-grasp tool perturbations.

IV. Method

The POGS system has 3 phases:

1) **Scene Capture** phase to obtain a set of images of a novel environment in a multi-view manner, maximizing different perspectives on objects of interest.
2) **Training** phase for aggregating and distilling information from captured views into a unified POGS representation.
3) **Persistent Object Tracking** phase for online tracking and updating of object poses as they move through the workspace from human or robot manipulation.

A. Scene Capture

The initial scene is scanned using an RGBD ZED Mini stereo camera mounted on a UR5 robot end effector. We capture images from 35 views as the robot moves along a predefined trajectory around the workspace to have sufficient viewpoints. Depth images are obtained from stereo pairs with RAFT-Stereo inference [46], which are deprojected into a fused pointcloud and used for initializing the Gaussian Splat. We run DBSCAN [47] on the fused pointcloud to filter noise and floater points.

B. Gaussian Splatting with Feature Fields

The goal of the training phase is to fuse information from the collected multi-view images and generate the unified 3D representation POGS for all objects in the scene. Rendering and supervising color for POGS remains exactly the same as 3DGS. We additionally regularize the 3D geometry with a depth reconstruction objective to encourage Gaussian means to be positioned on object surfaces [48]. We employ feature rendering techniques [20, 25] to simultaneously distill useful latent and explicit features from the 2D images into the 3DGS. In particular, we supervise into the 3DGS feature field:

Grouping Features: 2D object-level masks are supervised into 3D features for object clustering and singulating them from the environment. We obtain 2D masks from the training images using the object detection and segmentation network Detic [10].

To distill 2D object masks into 3D gaussian partitions, we borrow principles from [49, 50] and train a feature embedding encoder F_{emb} that passes an input gaussian mean position $\vec{x} \in \mathbb{R}^3$ through a hash-grid encoder [51] followed by an MLP, outputting a D-dimensional embedding vector. 3D gaussian features are rendered from a specific camera location to produce a feature map. For this we use Nerfstudio's [52, 53] 3DGS tile-based rasterizer implementation, with gradients backpropagated through the MLP within F_{emb}. 3D grouping features are then supervised with the contrastive objective from Bhalgat *et al.* [50], which operates through two complementary mechanisms: (1) attracting features that belong to the same object mask by minimizing their distance in embedding space, and (2) repelling features from different object masks by maximizing their embedding distances. We observe that computing and including a negative mask (wherever an object

mask does not exist) is helpful in reducing group feature noise for the scene background (anything in the scene that is not a tracked object).

Before the tracking phase begins, the system must identify and segment individual objects within the scene. As proposed in GARField [49], this is accomplished by clustering the group features using HDBSCAN. The result is a mapping from each 3D gaussian to a mask and label.

Language Features: To facilitate natural language queries in 3D, we incorporate multi-scale CLIP pyramid features distilled into the 3D gaussians, following the methodology described in [19] and [24]. Specifically, a scale-conditioned language feature embedding function is defined, $F_{lang}(\vec{x}, s) : (\mathbb{R}^3, \mathbb{R}) \rightarrow \mathbb{R}^D$ which maps a position $\vec{x}$ and physical scale s to a language-aligned embedding vector.

As in LERF, during deployment we use the CLIP text encoder to obtain embedding vectors for arbitrary natural language input queries. The relevancy of each gaussian to a given text query is computed by taking the cosine similarity score between the gaussian's language embedding and the text embedding.

Self-Supervised Features for Object Tracking: We distill dense visual features extracted from DINOv2 [9] into the 3D gaussians during training. These features are then supervised into the gaussians, enabling the model to render them at deployment time for optimizing object tracking objectives, similar to the method described in Robot See, Robot Do [54] and further detailed in the next sections.

Unlike the object grouping features and language features where we learn embedding functions to map inputs into feature space, the supervision of DINO visual features into POGS instead directly renders and optimizes trainable feature vectors of dimension-d with each Gaussian primitive. To integrate the DINO features efficiently, we apply principal component analysis (PCA) to reduce their dimensionality from several hundred to $d = 64$ dimensions. Without dimensionality reduction, storing per-Gaussian feature vectors would be computationally prohibitive.

We use Nerfstudio's [55] Splatfacto implementation of Gaussian Splatting with the gsplat [53] backend and modify it with the aforementioned image encoders and feature supervision losses.

C. Persistent Object Representation

Because POGS contains language, grouping, and visual features in a single representation, POGS can be used to query for objects with natural language and track those objects online by representing each object as a cluster of gaussian primitives in 3D. For each gaussian cluster, the system uses the centroid of gaussians within that cluster as the object frame, canonicalized such that the initial pose of each object is rotated to align with the world frame.

POGS extracts DINO visual features from live stereo camera observations, from only the left camera of the stereo pair. Feature rendering from the POGS model can directly obtain a synthetic view of the object feature maps from the same perspective as the real camera by using calibrated extrinsics. Simultaneously, POGS captures depth maps that serve as ground truth geometry to further regularize object pose. To generate accurate depth maps from stereo images, we employ a neural depth estimation model developed by Toyota Research Institute (TRI) [56], chosen for memory efficiency and real-time inference. This model operates effectively at an image resolution of 1080p, with depth inference frequency at approximately 30 Hz.

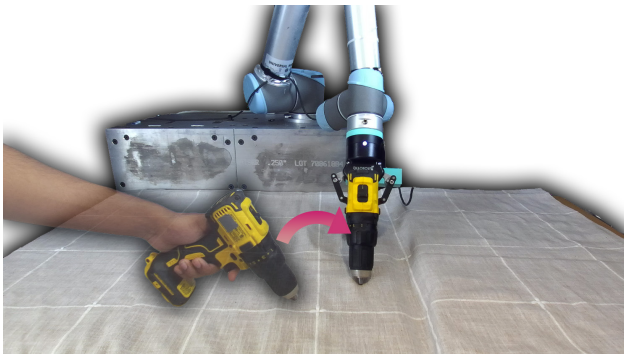

Fig. 3: **Occluded Grasp Sampling** POGS is capable of sampling and performing robot grasps on geometry that is fully occluded from the observation camera view (shown). The drill handle is fully occluded by the motor body, yet our POGS unified representation enables handle grasping based on previously observed geometry.

D. Tracking with POGS

Inspired by RSRD [54], the core of the tracking algorithm is the computation of the loss between the distilled DINO features of the rendered Gaussian Splat and the observed images. This feature loss measures how well the current pose estimates visually align the rendered model with the actual objects. POGS also includes a depth loss term that compares the rendered depth maps with depth maps extracted from the real observations, enforcing geometric consistency. The total loss is a weighted sum of the feature loss and depth loss, guiding the optimization to adjust per-object pose parameters until convergence. Each Gaussian cluster pose parameter is optimized independently, allowing POGS to track multiple moving objects, without imposing constraints on their relative movements. unlike prior work in real-time tracking of gaussian splats.

For each new frame captured by the camera, POGS repeats the rendering, feature extraction, loss computation, and optimization steps. This iterative process continually refines the pose estimates, improving alignment between the rendered clusters and the observed images over time.

E. Human & Robot Manipulation

We deploy POGS for tracking human and robot manipulation tasks where objects may be in varying poses compared to their initial positions in the scene capture. To facilitate robot grasping based on language queries, POGS first identifies the object cluster that corresponds to the query. The Gaussian means representing that object cluster are passed as a point cloud to Contact-GraspNet [57], which generates potential grasp candidates along with their respective scores, and the highest scored grasp is executed. By using the Gaussian

Fig. 4: **Object Reset Experimental Setup** *Middle:* A human randomly perturbs the configuration of the tracked objects according to the two tiers. *Right:* A robot arm then plans a grasp on language-queried objects and performs object reset. This process repeats until errors in object state estimation are too high to recover for grasping.

means, the object grasp is based on the full 3D object geometry embedded in POGS, which can be beneficial compared to methods that solely deproject depth and are partially occluded as seen in Figure 3.

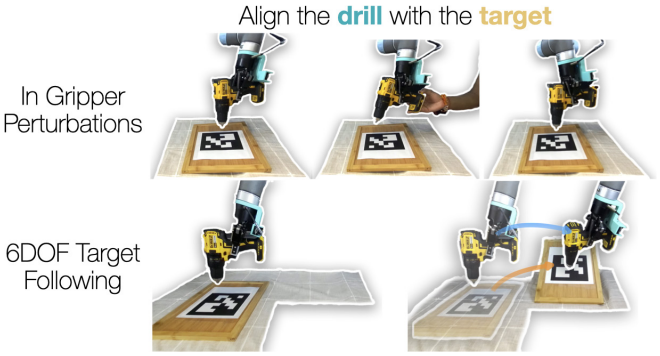

Fig. 5: **Tool Servoing Experimental Setup** The robot continuously attempts to align the tracked tool with the target. *Top:* A human perturbs the tracked tool while in the robot's gripper. The robot adjusts its end-effector position with closed-loop control to re-align the object with the target. *Bottom:* As a human shifts and rotates the target into new poses, the robot moves so the tool follows the target while maintaining alignment.

V. Physical Experiments

For physical experiments, we use a UR5 robotic arm with a static ZED 2 stereo camera. The POGS model is trained and initialized on a PC workstation with an NVIDIA 4090 GPU. We evaluate POGS on two robotic manipulation tasks across various objects. These tasks test POGS's ability to track objects of interest when manipulated by a robot or a human.

Both tasks begin with the UR5 using a wrist-mounted ZED-Mini stereo camera to scan a scene and initialize a POGS. Scene scanning with the predefined hemispherical trajectory takes on average 2 minutes and training a POGS takes on average 3 minutes.

A. Sequential Object Reset

This experiment evaluates POGS's localization accuracy in sequential object reset tasks guided by natural language. The tasks involve irregular objects of various shapes, sizes, and weights: a jigsaw, clothes iron, shoe, shelf, and shoebox.

Before each trial, a human randomly perturbs the positions and orientations of all objects, with perturbations defined in two tiers: In tier 1, objects could be translated anywhere within the ZED 2 camera frustum but rotated only up to 90° around the vertical axis from their initial configuration. In tier 2, objects could be translated anywhere within the frustum and rotated to any magnitude around the vertical axis.

The operator provides natural language instructions specifying which object to grasp and where to place it. The robot executes the planned grasp on the target object, adjusts its orientation in the gripper to align with the major axes of the target placement object, and moves the grasped object to the placement location. After each pick-and-place operation, the scene is reset by placing the grasp object back onto the tabletop, after which the human operator perturbs the objects. Tracking remains running the entire time, and these consecutive object resets continue until POGS loses tracking of the objects, defined as when repeated grasp planning failures occur due to irrecoverable errors in object state estimation.

We assess and report in Table I the performance across 3 pick objects and 2 place objects, conducting five trials per pick object on both tiers. The performance metrics included the maximum and mean number of consecutive successful object resets without losing tracking, the successful object reset rates, and the mean and standard deviation of the translation error between the intended and actual placement positions. For example, in the "Clothes Iron to Shelf" task under Tier 1, POGS achieved a maximum of 12 consecutive successful object resets, with a successful pick rate of 32 out of 36 attempts and a mean translation error of 3.4 cm measured by calipers on reference markers made to each object. Under Tier 2, despite more extreme perturbations including full object rotations, POGS achieves 6 consecutive operations, and a mean translation error of 2.2 cm. Similar performance trends were observed in the other tasks, where POGS consistently outperformed ablations that either had depth perception turned off or were optimized with RGB substituting for DINO features. The ablations highlight the critical role that both depth perception and robust visual features play in achieving accurate object localization and successful sequential object resets.

B. Tool Servoing

We consider a tabletop workspace with a tool object and an ArUco marker fixed to a target object surface. In these experiments, the tool object is a drill and the target object is a wooden platform. The tool is manually annotated with a coordinate frame at the tool tip to indicate which component to align with the target (i.e. align the drill tip perpendicularly). The robot then grasps the tool and aligns it to the tracked ArUco marker. The robot then performs 6DoF visual servoing

	Jigsaw to Shelf				Clothes Iron to Shelf				Shoe to Shoerack			
	Tier 1			Tier 2	Tier 1			Tier 2	Tier 1			Tier 2
	No Depth	No DINO	POGS	POGS	No Depth	No DINO	POGS	POGS	No Depth	No DINO	POGS	POGS
Max Consecutive OR	1	0	**11**	3	3	0	**12**	6	2	0	**8**	7
Mean Consecutive OR	0.33	0	**4.4**	1.6	0.6	0	**7.2**	3	1.0	0	**6.4**	3.8
Successful Pick Rate	2/4	0/3	**23/27**	9/16	6/9	0/3	**32/36**	15/18	7/10	0/3	**34/38**	20/24
Successful Place Rate	1/2	-	**22/23**	8/9	3/6	-	**28/32**	13/15	5/7	-	**32/34**	19/20
Mean Position Error (cm)	2.1	-	**2.5**	1.8	7.0	-	**3.4**	2.2	4.7	-	**4.1**	3.5
Std Position Error (cm)	0.0	-	**1.0**	1.1	3.7	-	**2.6**	0.8	1.5	-	**1.5**	1.3

TABLE I: **Object Reset Results** *Consecutive OR* refers to a single trial with repeated Object Resets without losing tracking out of 5 trials for main experiments, and 3 trials for ablations. Note that the denominators for the successful pick rate and successful place rate metrics vary across trials. This variation arises because each trial was executed until a grasping failure occurred—i.e., when the error in object state estimation became too high to recover—resulting in a different total number of reset attempts per trial.

	Tier 1		Tier 2	
Perturbations	**Success Rate**	**Time (s)**	**Success Rate**	**Time (s)**
Clockwise	24/25	6.30	20/25	12.26
CCW	24/25	5.72	20/25	13.06
Follow Target	24/25	-	21/25	-

TABLE II: **Tool Servoing Results** We record results across 5 trials for each tier where each trial has the target object move to 5 poses and at each pose, the tool is perturbed clockwise and counter clockwise by 15-30 °. For the perturbations, we measure the time it takes for the drill to recover from the perturbation and realign with the target. Since the target was constantly moving over time, recovery time wasn't recorded for the follow target experiment.

such that the tool tip remains in its relative target pose to the object. We leave tool rotation about its target axis (i.e drill bit axis) unconstrained, and pick the orientation during servoing which minimizes tool motion. During each trial, the target object is moved to 5 random poses in the workspace and we record the success rate for how often the tool follows the object. At each pose, a human moves the drill 15-30° clockwise or counterclockwise about the grasp axis, and records how often the robot can adapt to this perturbation and locate the drill tip within 3cm and 5° of the target. We also report the average time taken for the robot to adapt to the perturbed gripper pose.

We run this experiment on two tiers with 5 trials each tier. Tier 1 experiments have the target object stay on the tabletop plane and can move anywhere within a 55 cm by 50 cm square, and the tool orientation in-grasp can be changed up to 15 degrees. Tier 2 experiments have the target moving in 3D space where it can move anywhere within a 55 cm by 72 cm by 10 cm box such that the ArUco marker was visible to the camera and the robot joints did not occlude the ArUco marker from the camera.

The results are reported in Table II. Overall, POGS can be used to recover from tool perturbances in gripper up to 15° in 48 of 50 trials at an average of 6.01 seconds. When the tool-in-gripper rotation increased to 30°, the success rate drops to 40 of 50 trials at a longer time average of 12.66 seconds largely because higher object deltas are harder to track. Similarly, for 2D space, the tool was able to follow the target object 24 of 25 trials but in 3D space that went down to 21 of 25 trials.

VI. Limitations

One key limitation of this work is that the online tracking frequency is limited to 5Hz on an NVIDIA 4090 GPU due to computational bottlenecks. This includes approximately 140ms latency for DINO feature extraction using a ViT-S, and multiple steps of optimization necessary per frame iteration to adjust the pose parameters of each object until convergence. As a result, objects had to be moved slowly with no sudden motions or quick changes in direction to avoid losing track of them. In future work, we will parallelize depth inference and DINO feature extraction for tracking speed optimization.

Another limitation is that objects that are partially occluded (by a hand, a robot gripper, etc.) have less robust tracking compared to fully unobstructed objects due to degraded tracking feature alignment between the real features and rendered features. In future work, we will add an end-effector regularization term for objects grasped by the robot as this is a useful prior constraining where the object can be in the workspace. Furthermore, we can also develop robot gripper masking to increase the alignment between rendered features and real features.

VII. Conclusion

In this work, we present Persistent Object Gaussian Splat (POGS), a system for tracking and manipulating irregularly shaped, previously unseen objects in dynamic environments. By integrating language, grouping, and self-supervised visual features into an explicit 3D Gaussian representation, POGS aims to address some of the challenges associated with CAD-based, NeRF-based, and conventional point cloud methods. Our experimental results suggest that POGS can maintain object state estimates during tasks such as object resets and tool servoing.

References

[1] 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, Jul. 2023.

[2] Y. Zhang, T. Wang, and Y. Zhang, "Tracking with the cad model of object for visual servoing," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2019.

[3] H. Wuest and D. Stricker, "Tracking of industrial objects by using cad models," *JVRB-Journal of Virtual Reality and Broadcasting*, vol. 4, no. 1, 2007.

[4] C. Wiedemann, M. Ulrich, and C. Steger, "Recognition and tracking of 3d objects," in *Joint Pattern Recognition Symposium*, Springer, 2008, pp. 132–141.

[5] O. Zhou, Y. Ge, Z. Dawei, and Z. Zhonglong, "A survey of rgb-depth object tracking," *Computer-Aided Design & Computer Graphics*, 2024.

[6] W.-L. Zheng, S.-C. Shen, and B.-L. Lu, "Online depth image-based object tracking with sparse representation and object detection," *Neural Processing Letters*, vol. 45, pp. 745–758, 2017.

[7] 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," *ACM*, vol. 65, no. 1, 2021.

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

[9] M. Oquab *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[10] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision*, Springer, 2022, pp. 350–368.

[11] 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," in *7th Annual Conference on Robot Learning*, 2023.

[12] 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.

[13] B. Chen *et al.*, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11 509–11 522.

[14] 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.

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

[16] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[17] Q. Gu *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 5021–5028.

[18] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

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

[20] 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.

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

[22] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning*, 2022.

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

[24] J. Yu *et al.*, "Incrementally Building Room-Scale Language-Embedded Gaussian Splats (LEGS) with a Mobile Robot," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[25] S. Zhou *et al.*, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 676–21 685.

[26] R.-Z. Qiu, G. Yang, W. Zeng, and X. Wang, "Language-driven physics-based scene synthesis and editing via feature splatting," in *European Conference on Computer Vision (ECCV)*, 2024.

[27] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.

[28] Y. Zheng *et al.*, "Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping," *IEEE Robotics and Automation Letters*, 2024.

[29] B. Wen and K. Bekris, "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021.

[30] B. Wen *et al.*, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[31] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[32] J. Sun *et al.*, "Onepose: One-shot object pose estimation without cad models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6825–6834.

[33] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians:tracking by persistent dynamic view synthesis," in *3DV*, 2024.

[34] G. Wu *et al.*, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[35] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 20 331–20 341.

[36] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conference on Robot Learning*, PMLR, 2018, pp. 373–385.

[37] W. Gao and R. Tedrake, "Kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[38] A. Simeonov *et al.*, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 6394–6400.

[39] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning dense object descriptors from neural radiance fields," in *IEEE Conference on Robotics and Automation (ICRA)*, 2022.

[40] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *8th Annual Conference on Robot Learning*, 2024.

[41] Y. Wang *et al.*, "D^3 fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement," in *8th Annual Conference on Robot Learning*, 2024.

[42] O. Shorinwa *et al.*, "Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting," in *8th Annual Conference on Robot Learning*, 2024.

[43] Y. Li and D. Pathak, "Object-aware gaussian splatting for robotic manipulation," in *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[44] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, "Graspsplats: Efficient manipulation with 3d feature splatting," in *8th Annual Conference on Robot Learning*, 2024.

[45] J. Kerr *et al.*, "Robot see robot do: Part-centric feature fields for visual imitation of articulated objects," in *8th Annual Conference on Robot Learning*, 2024.

[46] L. Lipson, Z. Teed, and J. Deng, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *2021 International Conference on 3D Vision (3DV)*, IEEE, 2021, pp. 218–227.

[47] 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*, vol. 96, 1996, pp. 226–231.

[48] J. Chung, J. Oh, and K. M. Lee, "Depth-regularized optimization for 3d gaussian splatting in few-shot images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 811–820.

[49] C. M. Kim, M. Wu, J. Kerr, M. Tancik, K. Goldberg, and A. Kanazawa, "Garfield: Group anything with radiance fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[50] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi, "Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[51] 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.

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

[53] V. Ye *et al.*, "Gsplat: An open-source library for gaussian splatting," *arXiv preprint arXiv:2409.06765*, 2024.

[54] J. Kerr *et al.*, "Robot see robot do: Part-centric feature fields for visual imitation of articulated objects," in *8th Annual Conference on Robot Learning*, 2024.

[55] M. Tancik *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[56] 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.

[57] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 438–13 444.