Object-Aware Gaussian Splatting for Robotic Manipulation

Yulong Li and Deepak Pathak

Carnegie Mellon University

***Abstract*— Understanding the dynamics of our world in 3D is critical for the performance and robustness of robotics applications. Although recent progress has married vision foundation models and volumetric rendering to offer semantic 3D representations, neither the inference time of large models nor the update speed of volumetric representation meets the desired update rate of real-time robotic manipulation. In this work, we propose to inject "objectness" into a semantic representation based on 3D Gaussians [1]. The Gaussians with the same semantic labels can initialize and update together, leading to fast updates in response to robot and object movements. All necessary semantic information is extracted at the initial step from pretrained foundation models, thus circumventing the inference bottleneck of large models but still obtaining semantic information. With only three camera views, our proposed representation is able to capture a dynamic scene at 30 Hz in real-time, which is sufficient for most manipulation tasks. Leveraging the representation based on our object-aware Gaussian splatting, we are able to solve language-conditioned dynamic grasping, for which the robot grasps dynamically moving objects specified by open vocabulary queries. We also use the representation to train a visuomotor policy via behavior cloning and show that the policy achieves comparable results with image-based policies with pretrained encoders. Videos at https://object-aware-gaussian.github.io**

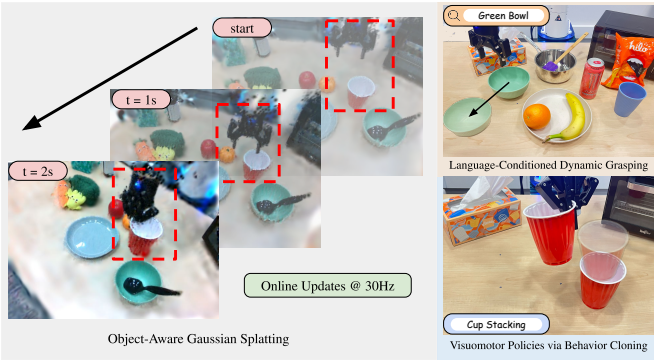

Fig. 1: **Object-aware Gaussian splatting.** We propose a dynamic and semantic 3D representation based on Gaussian Splatting [1]. The representation achieves an update rate of 30 Hz, and faithfully represents robot and object movements with only three training views. We apply this representation to zero-shot language-conditioned dynamic grasping and demonstrate its applicability to visuomotor policy training.

I. Introduction

A pivotal element in robotic manipulation is the representation of the scene. While 2D images are readily accessible and significantly benefit from advancements in vision foundation models [2], [3], [4], [5], they lack the essential 3D understanding required for complex robotic tasks. Recent strides in integrating semantic information into neural 3D representations, such as Neural Radiance Fields (NeRF)[6], have shown promise in enabling tasks like language-conditioned grasping[7], [8]. Yet, these approaches stumble when faced with dynamic scenes—an essential aspect of robotic applications.

The crux of the challenge lies in the resource-intensive demands of constructing semantic 3D representations which are already compute and memory-intensive for passive vision applications. But robotics adds an additional axis of action, making the representation space $x \times y \times z \times t$ where t is the number of steps required to execute a task which easily scales to 100s-1000s in the simplest of tasks as robots need to be controlled at 10Hz frequency at least. This makes 3D representation for robotics exponentially more demanding, combined with the fact that this needs to be real time which is an indispensable requirement for the dynamic world of robotic manipulation.

However, a close examination of the robotic tasks reveals a potential solution. Changes within a scene between updates are predominantly localized, suggesting that a per-step scene reconstruction may not only be inefficient but also unnecessary. By transitioning to a locally updatable scene representation, we can directly address the core of the computational challenge. This pivot from continuous, global reconstruction towards targeted, localized updates dramatically curtails the overhead associated with keeping a semantic and dynamic 3D representation, where the main computation is completed at the initialization.

Gaussian splatting [1] emerges as a promising candidate for dynamic 3D scene representation in this context. Originating from novel-view synthesis, this method employs a set of 3D Gaussian primitives to model a scene. This explicit and volumetric representation allows for local updates of the constructed scene. Further, its reliance on rasterization for rendering leverages parallel processing on GPUs, markedly accelerating rendering speeds. Nonetheless, adapting Gaussian splatting for robotics poses its own set of challenges. While it offers a speed advantage, it lacks the semantic understanding of the scene, and vitally, it still falls short of meeting the real-time update requirements for robotics.

In response to these challenges, our work builds upon static Gaussian splatting to bridge this gap. We address the need for speed and semantic interpretation by embedding "objectness" into the scene representation, thereby expediting the update process. This approach allows for rapid, high-frequency updates essential for dynamic robotic environments. This also allows a one-time extraction of 2D foun-

dation models at the initial step for semantic information, circumventing the inference bottleneck of large models.

We demonstrate the practicality of our method through the task of language-conditioned dynamic grasping. In this scenario, a robot employs our proposed representation to reactively grasp moving objects prompted by open-vocabulary queries. We also showcase the potential of the representation by integrating it into a visuomotor policy trained through behavior cloning.

We believe that the scene representation holds significant potential for a wide range of applications within the field of robotics.

II. Dynamic Object-aware Gaussians

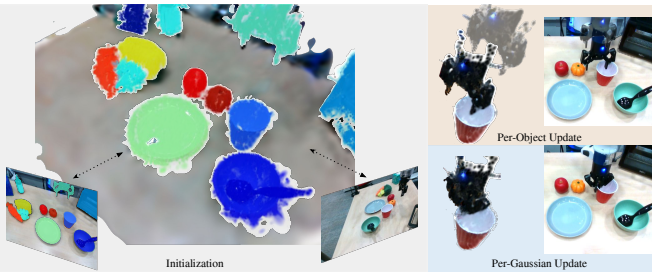

Fig. 2: **Method Overview.** We obtain object-wise segmentation from 2D foundation models [4] at initial reconstruction. In the following updates, objects are first initialized to an approximate 3D position, and then the displacement is optimized with photometric loss. We also optimize for the displacements of individual Gaussians to account for non-rigid transformations like the closing of the robot gripper.

A. Problem Formulation and Initial Reconstruction

We seek to construct a semantic and dynamic 3D representation S_t of the scene for each step t given views from a few RGB-D cameras. For each camera c, we have the data tuple $(I_{c,t}, D_{c,t}, E_{c,t}, K_c)$, where $I_{c,t}$ is the RGB image, $D_{c,t}$ is the depth image, $E_{c,t}$ represents the time-dependent camera extrinsic, and K_c denotes the camera intrinsic. These cameras may be static, affixed to the robot or other moving objects. Our main challenge is to update the scene at a high frequency (30 Hz).

Due to the requirement for update speed and limited camera views in robotic applications, relying solely on spatial information from the current time step is inadequate for accurate reconstruction. Our proposed solution seeks not only to reconstruct the scene S_t using spatial information but also to enrich it with temporal information from previous time steps. This is achieved by auto-regressively reconstructing S_t from S_{t-1}, thereby implicitly utilizing information from all previous time steps. By doing this, the scene representation also naturally exhibits temporal continuity, possibly allowing the agent to capture and reflect changes over time. This also allows the computations, such as semantic extractions, at the initial time step to be carried over.

We propose to use the 3D Gaussians [1] as our scene representation: S_t is represented by a set of 3D Gaussians, $(\mathbf{x}_{i,t}, \mathbf{R}_i, \mathbf{s}_i, \mathbf{c}_i, \alpha_i)$, where the Gaussian centers are time-variant. At the initial time step, we initialize the scene with a dense point cloud obtained from the camera views. This ensures the initial reconstruction is regularized even though the views are few. We also obtain semantic features relevant to the task from 2D foundation models.

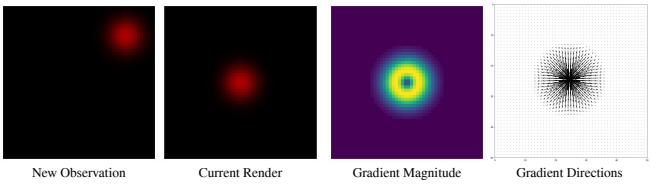

Fig. 3: **Locality of Photometric Loss**: Given a new observation where the ground truth center of the Gaussian moves far away, the gradients on the Gaussian centers become uninformative of the desired position.

Upon obtaining the initial scene S_0, a naive approach for progressing to S_1 involves using the spatial parameters of S_0 as initial values for $\mathbf{x}_{i,1}$, and then refining these parameters with new observations $(I_{c,1}, E_{c,1}, K_c)$. This method, however, faces two primary issues: limited camera views at subsequent time steps can lead to overfitting, such as moving excess points from the background to incorrectly cover moving foreground objects; and the approach is too slow for the rapid updates required in robotics. To address these challenges, we introduce object-aware initialization and updates, as illustrated in Fig. 2.

Incorporating objectness into the Gaussian scene representation is a pivotal aspect of our method. Besides reconstructing the geometric scene with 3D Gaussian Splatting, the initial step in our approach also utilizes pretrained segmentation models to obtain instance segmentation of the scene. Specifically, we pick one camera view and its associated RGB image I_c, and obtain a segmentation mask M_c. The segmentation labels are then lifted into 3D space through camera matrices and depth D_c, so that each point in the point-cloud extracted, $\mathscr{P}_c$, has a corresponding segmentation label. Finally, the point clouds obtained from other views inherit their respective segmentation labels from their nearest neighbors in $\mathscr{P}_c$. Thus, each 3D Gaussian is enhanced with a segmentation label k, $g_i = (\mathbf{x}_{i,t}, \mathbf{R}_i, \mathbf{s}_i, \mathbf{c}_i, \alpha_i, l_i)$, where $l_i \in \{1, \ldots, K\}$ for K detected objects. We further label the background with $l_i = 0$. In theory, many off-the-shelf segmenters is applicable for our purpose, but we obtain the segmentation map through GroundedSAM [4], [9], [10], [2], [5] with the language query "object". In the following sections, we introduce how to use the segmentation information to rapidly update the scene given dynamic movements.

B. Object-centric Initialization

Even though photometric loss provides great supervision to construct detailed and precise static scenes, it crucially relies on a good initialization of the 3D Gaussians. This is because the loss is inherently local for two main reasons. The scale of each Gaussian is limited and influences limited volume, and thus gradients vanish quickly if the Gaussian is initialized far away from the ground truth position. Further, to allow for parallel rendering through rasterization, the image is split into tiles and gradients cannot propagate across tiles. In fact, even if the gradients propagate without vanishing, due

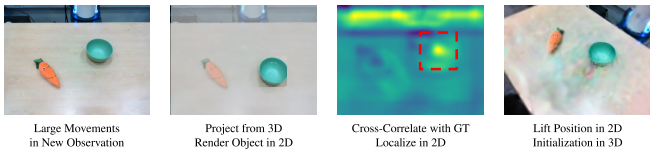

Fig. 4: **Spatial-temporal representation with Gaussian splatting**: The locality of photometric loss prevents the Gaussians to move to the desired position if the movement between updates is too large. to the non-convex optimization landscape, the Gaussians are likely to stuck in local minima, like moving objects below the table in a table-top setting.

However, in real-world tasks, the object movements between updates can be large, and thus solely relying upon the photometric loss can be limited. Therefore, we propose to use object-centric initialization through template-matching [11] to bootstrap the optimization. Given a camera view $(I_{c,t}, E_{c,t}, K_c)$ at time step t, we obtain the rendering $I^{\text{render}}_{c,t-1}$ from the reconstructed scene S_{t-1} from the view. The rendering is off due to movements at step t, but by comparing $I'_{c,t-1}$ and I_t, we will have a good guess of how the objects move. This process involves three steps. The first step is to localize an object and extract the template. For each object k, we render the scene where only the Gaussians with labels $l_i = k$ are visible from view c to obtain the center of the object k, $p_{k,t-1} = (u_k, v_k)$, and obtain a local crop of side length $2l$ around the center

$$T_n = I^{\text{render}}_{c,t-1}[u_n - l : u_n + l, v_n - l, v_n + l]$$

as the template. Then we match the template with the current observation I_t to obtain the new center $p_{n,t}$. Specifically, we use cross-correlation for template matching which can be efficiently computed on GPUs. Finally, the new center $p_{n,t}$ is lifted to ray in 3D using the camera matrices. By finding the intersection of the rays from two views or utilizing the depth image $D_{c,t}$, we obtain an initialization of the new 3D center of the object.

C. Object-centric Updates

The object-centric initialization helps overcome the locality of the photometric loss. We then optimize the displacements of each Gaussian to minimize the photometric loss. However, optimizing each individual Gaussians freely can still lead to overfitting or nonphysical deformation of objects due to limited views and few number of updates. To regularize the update, we introduce D_k as the group displacement for each object k. We also introduce an individual displacement δ_i for each Gaussian g_i to account for rotations and non-rigid transforms such as the closing of the robot gripper. At each step, D_k is initialized as discussed in the previous section, and δ_i is initialized with the value from the previous step to carry on the momentum in the transformation.

Finally, an essential modification is made for background Gaussians (labeled $l_i = 0$), which are kept fixed during optimization. This constraint is instrumental in preventing the model from overfitting by relocating background Gaussians to improperly occlude or merge with foreground objects. It ensures that the background remains stable and consistent across updates, thereby focusing the optimization process on accurately capturing and tracking the movement and deformation of objects within the scene.

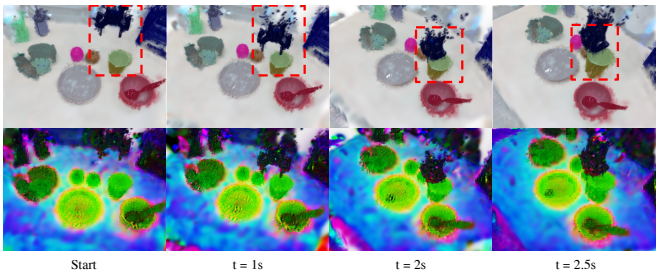

Fig. 5: **Dynamic Segmentation and Feature Maps.** We show the segmentation and feature map distilled from DINOv2 [5] at different time steps and different views.

III. Application to Robotics

A. Dynamic Segmentation and Feature Field

With our formulation, we can naturally enrich each Gaussian with semantic features, $g_i = (\mathbf{x}_i, \mathbf{R}_i, \mathbf{s}_i, \mathbf{c}_i, \alpha_i, l_i, f_i)$, which will be carried on through the real-time updates. The semantic features can be extracted from 2D foundation models through two approaches. If the dense per pixel 2D features are available, the straightforward way is to directly project from 2D space to 3D space at initialization, like the segmentation labels. Alternatively, we can extract 2D features from foundation models and distill the features into the Gaussians through optimization.

The representation makes possible online dynamic segmentation and feature extractions, as shown in Fig. 5.

B. Zero-shot Language-conditioned Dynamic Grasping

Our representation is readily applicable to zero-shot language-conditioned dynamic grasping. In this setting, a user issues a language query for the robot to grasp a specified object without prior demonstrations. The task is complicated by the possibility that the target object may be moving, requiring the agent to adapt dynamically. At the initialization stage, we extract a language-aligned feature f_k for each object k with CLIP [3]. Then, at query time, the user's query q is matched with the closest object query based on cosine distance. At time t, we collect the centers of Gaussians marked by $l_i = k_q$, denoted as $\mathscr{P}_q$. This collection forms the basis for determining a viable grasp, parameterized by a pose T_t. In particular, we randomly sample grasp poses near the point-cloud $\mathscr{P}_q$ and take the grasp with the maximal antipodal score. A motion planner is then used to direct the robot to the pose specified by T_t.

C. Representation for Behavior Cloning Policies

Finally, we can use our representation to train visuomotor policies through behavior cloning. Since the representation is 3D, semantic, and dynamic, it has great potential in training sample-efficient behavior cloning policies.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–14, 2023.

[2] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 8748–8763, 2021.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

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

[6] 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.

[7] S. Sharma, A. Rashid, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023.

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

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

[10] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[11] J. P. Lewis, "Fast template matching," in *Vision interface*, vol. 95. Quebec City, QC, Canada, 1995, pp. 15–19.