3D-Aware Manipulation with Object-Centric Gaussian Splatting

Anonymous Author(s)
Affiliation
Address
email

Abstract:
3D Understanding of the environment is critical for the robustness and performance of robot learning systems. As an example, 2D image-based policies can easily fail due to a slight change in camera viewpoints. However, when constructing a 3D representation, previous approaches often either sacrifice the rich semantic abilities of 2D models or settles for a slower update rate that hinders real-time robotic manipulation. In this work, we propose a 3D representation based on 3D Gaussians [1] that is both semantic and dynamic. With only a single or a few camera views, our proposed representation is able to capture a dynamic scene at 30 Hz in real-time in response to robot and object movements, which is sufficient for most manipulation tasks. Our key insight in achieving this fast update frequency is to make object-centric updates to the representation. Semantic information can be extracted at the initial step from pretrained foundation models, thus circumventing the inference bottleneck of large models during policy rollouts. Leveraging our object-centric Gaussian representation, we demonstrate a straightforward yet effective way to achieve view-robustness for visuomotor policies. Our representation also enables language-conditioned dynamic grasping, for which the robot perform geometric grasp of moving objects specified by open vocabulary queries. Please refer to https://object-aware-gaussian.github.io for more results.

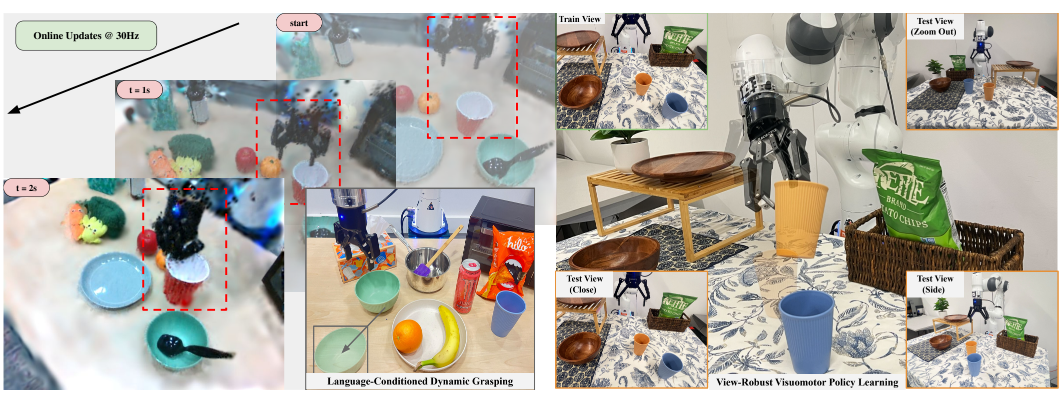


Figure 1: **Object-centric Gaussian splatting.** We propose a dynamic and semantic 3D representation based on Gaussian Splatting [1], which achieves an update rate of 30 Hz in response to robot and object movements. We show the reconstruction from different viewpoints of a grasping scene on the left. We apply this representation to obtain behavior cloning policies that are robust under various testing views even though only a single training view is available. We also apply our representation to enable zero-shot language-conditioned dynamic grasping.

1 Introduction

What representation of the scene will improve the performance and robustness of learning robots? Recent achievements in the community suggest that taking 2D RGB images as inputs allow robots to

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

perform complex manipulation tasks [2, 3]. Nevertheless, the hidden assumption is that the camera viewpoints remain the same for training and testing. As we will demonstrate in Sec. 4.1, even slight shift in camera views will significantly reduce the performance of learning agents. A fixed relative pose between the cameras and the robot base or the end-effectors is an unsatisfactory requirement. As humans, we can easily solve the same tasks without our eyes fixing at a position relative to our hands. We can even easily tele-operate a robot to complete the task at completely different views. Unfortunately, most of the existing learning agents lack the 3D understanding essential to robustness of the policies.

There has been promising results on directly learning with 3D representations like voxels or point-clouds [4, 5], yet it would be optimal if learning agents can leverage immense 2D data and readily accessible pretrained vision foundation models [6, 7, 8, 9, 10]. Recent strides in integrating semantic information into neural 3D representations [11] have shown promise in enabling tasks like language-conditioned grasping [12, 13] and goal-conditioned rearrangement [14]. Yet, these approaches stumble when faced with dynamic scenes and the requirement of higher-frequency (30Hz) controls, constraining their general applicability.

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. Robotics adds an additional axis of time, requiring controllers at 10Hz frequency at least for practical applications. The indispensable requirement for real-time updates of the dynamic world makes 3D representation for robotics exponentially more demanding.

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

With our representation, we can robustify off-the-shelf 2D policy trainers to handle arbitrary camera poses by projecting observations to training views. Our semantic, dynamic, and 3D representation also allows a robot to reactively grasp moving objects prompted by open-vocabulary queries.

In summary, our contributions are:

1. Introducing the use of object-centric Gaussian splatting for dynamic, semantic, and 3D representation in robotics.
2. Overcoming the update speed limitations of the vanilla Gaussian splatting through object-centric updates, achieving 30 Hz update rate which is sufficient for most real-time robotic applications.
3. Proposing GSMimic, which utilizes our representation to obtain view-robust behavior cloning policies evaluated on simulation and real-world manipulation tasks.
4. Demonstrate the representations applicability to zero-shot language-conditioned dynamic grasping, showcasing its adaptability in dynamic settings.

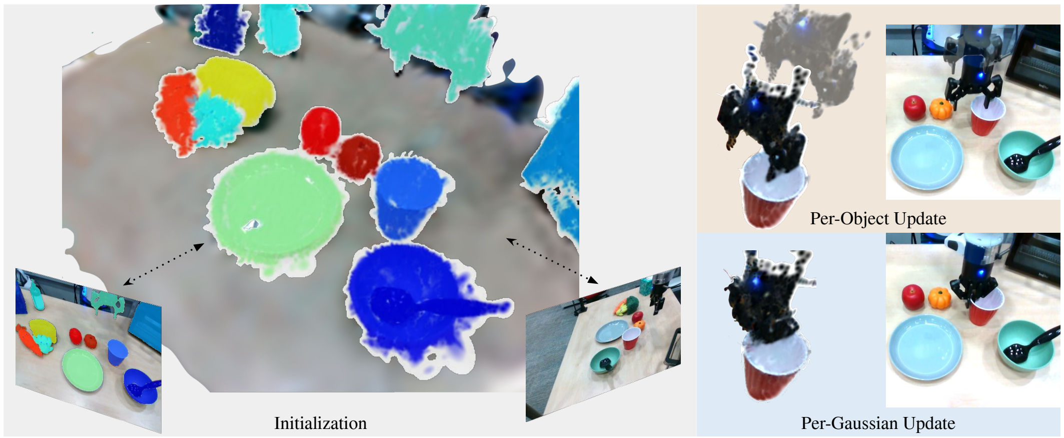


Figure 2: **Method Overview.** We obtain object-wise segmentation from 2D foundation models [8] at initial reconstruction. In the following updates, objects displacements are optimized with photo-metric loss. We also optimize for the displacements of individual Gaussians to account for non-rigid transformations like the closing of the robot gripper.

2 Dynamic Object-centric Gaussians

2.1 Preliminaries on Gaussian Splatting

Our initial scene representation is constructed based on 3D Gaussian Splatting [1]. The scene is represented by a collection of 3D Gaussians, where the ith Gaussian is specified by a set of learning parameters: $\mathbf{x}_i \in \mathbb{R}^3$ is Gaussian center, $\mathbf{R}_i \in SO(3)$ the rotation, $\mathbf{s}_i \in \mathbb{R}^3$ the scale, $\mathbf{c}_i \in \mathbb{R}^3$ the color, and $\alpha_i \in \mathbb{R}$ the opacity. The weight w_i of each g_i on a point $\mathbf{p}$ in 3D space is determined by the Gaussian distribution, adjusted by the opacity:

$$w_i(\mathbf{p}) = \sigma(\alpha_i) \exp\left(-\frac{1}{2}(\mathbf{p}-\mathbf{x}_i)^\top \mathbf{\Sigma}_i^{-1}(\mathbf{p}-\mathbf{x}_i)\right)$$

where $\sigma(\cdot)$ denotes the sigmoid function, and $\mathbf{\Sigma}_i$ is the covariance matrix, derived from its rotation and scale. To render an image I^{render} from a camera viewpoint, the 2D center of a Gaussian g_i is projected onto the image plane using the camera matrices. The 2D weight w_i^{2D} is similarly computed with the 2D center and the covariance. All the 2D centers are sorted then by depth in ascending order, and pixel color $I^{\text{render}}[u, v]$ is accumulated:

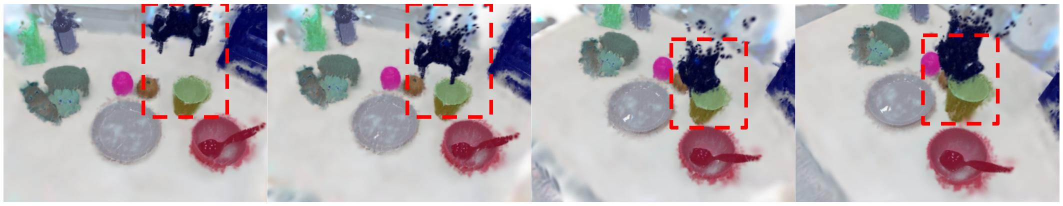
$$I^{\text{render}}[u, v] = \sum_i \mathbf{c}_i w_i^{\text{2D}}(u, v) \prod_{j=1}^{i-1}(1 - w_j^{\text{2D}}(u, v))$$

Finally, given a ground-truth image I from the viewpoint, the Gaussian parameters can be optimized by minimizing a differentiable photometric loss that measures that distance between I and I^{render}. This optimization process is fully differentiable and designed for GPU-based parallel computation, ensuring rapid training.

2.2 Problem Formulation and Initial Reconstruction

We seek to construct a semantic and dynamic 3D representation S_t of the scene for each time step t given views from a few RGB-D cameras. For each camera labeled with 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.



Start | t = 1s | t = 2s | t = 2.5s

Figure 3: **Dynamic Segmentation.** We show the segmentation map at different time steps and rendered at different views.

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.

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 updating 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-centric 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, $\mathcal{P}_c$, has a corresponding segmentation label. Finally, the point clouds obtained from other views inherit their respective segmentation labels from their nearest neighbors in $\mathcal{P}_c$. 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 $l_i = 0$. We visualize this initial segmentation on the left of Fig. 2, and this segmentation is carried on in the following dynamic updates, as shown in Fig. 3. In theory, many off-the-shelf segmenters is applicable for our purpose, but we obtain the segmentation map through GroundedSAM [8, 15, 16, 6, 9] 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.

2.3 Object-centric Updates

Optimizing each individual Gaussians freely can lead to overfitting or nonphysical deformation of objects due to limited views and few number of updates. To regularize the update, we introduce G_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 a step t, G_k is initialized with the value obtained at step $t-1$ to carry over some momentum, and δ_i is initialized with zeros.

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. We summarize the pipeline in Algorithm 1. Our method achieves update rates of up to 30Hz, aligning with the dynamic needs of robotic operations.

Algorithm 1 Dynamic Gaussian Splatting for Real-time Robotics

Require: $n_{\text{step}} = 3$
  for time step t **do**
    Set $\delta_i := 0$ for each Gaussian i where $l_i \neq 0$
    Receive camera views $V_t = \{(I_{c,t}, E_{c,t}, K_c)\}$
    if $t = 0$ **then**
      S_0, K := Initialize(V_t)
      Set $G_k := 0$ for each object k
    else
      for step in n_{step} **do**
        $x_{i,t} := x_{i,t-1} + G_k + \delta_i$ for $l_i = k$, for $k \in \{1, \ldots, K\}$
        Render I_c^{render} and compute loss $\mathcal{L}_c$
        Perform gradient updates: $G_k := G_k - \alpha_0 \nabla_{G_k} L_c$, $\delta_i := \delta_i - \alpha_1 \nabla_{\delta_i} \mathcal{L}_c$
      end for
    end if
  end for

3 3D-Aware Manipulation

To demonstrate the usefulness of our representation, we propose two straightforward yet effective applications of our representation to robotic manipulation. First, we show how to achieve view-robustness for image-based visuomotor policies. Second, we applies our representation to enable grasping of moving unseen objects conditioned on open-vocabulary language queries.

3.1 View-Robust Visuomotor Policy Learning via GSMimic

Consider a visuomotor policy which takes as inputs RGB images from a set of cameras. The problem of view-robustness arises if the training viewpoints are fixed to a coordinate frame, for example, the world frame or the end-effector frame. If the cameras are mounted differently during training time, the changes in input observation create a distribution shift that leads to significant performance drop. This issue cannot easily be handled during training without additional training cameras. With object-centric Gaussian representation, we can circumvent this issue with the additional depth input. During test-time, we can render via our 3D scene representation to get pseudo observations from the same viewpoints as training time. One of the complications is that due to limited field-of-view, test-time viewpoints will not fully cover the training viewpoints, creating empty areas in the rendering. To fix this, we directly train with renderings of foreground Gaussians only by removing Gaussians with label $l_i = 0$ during rendering. We specifically evaluate this strategy on visuomotor policies trained via behavior cloning, and term the overall approach GSMimic.

3.2 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 $\mathbf{f}_k$ for each object k with CLIP [7]. Then, at query time, we use CLIP to extract an embedding $\mathbf{f}_q$ for the query, and the query is matched with the objects in the scene based on cosine distance:

$$k_q = \underset{k \in \{1, \ldots, K\}}{\arg\max} \frac{\mathbf{f}_k \cdot \mathbf{f}_q}{||\mathbf{f}_k|| \cdot ||\mathbf{q}||}$$

With the benefit of explicity 3D representation, at time step t, we are able to extract the point-cloud of the target object $\mathcal{P}_q$ by collecting the centers of Gaussians marked by $l_i = k_q$. The point-cloud 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 $\mathcal{P}_q$ 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. Both the semantics, dynamics, and 3D aspects are crucial for the sucess of the task.

4 Evaluation

4.1 View-Robust Behavior Cloning

In our experimental evaluation, we seek to investigate the generalization ability of GSMimic to unseen camera viewpoints during test time.

Simulation Evaluation. We used Robomimic [17], a large-scale robotic manipulation benchmark as our simulation testbed. We evaluated on the 4 single-arm Franka tasks from the benchmark: Lift, Can, Square, and Tool Hang. We used proficient human teleoperated demonstration dataset for each task, and use the RGB-D observation from the default "agentview" camera for the training.

Real-world Evaluation. We designed 2 tasks for real world validation on a Franka Panda Robot. (1) Cup Stacking requires the robot to pick up one of the cups on the table and place it into the other cup. (2) Cup Unstacking requires the robot to grasp the thin edge of the top cup, place it on the table, and then push it forward to roughly align with the other cup. Both tasks use Cartesian velocity control as the control space, and a proprioceptive inputs and a single front camera view as the observation space. We collect 50 tele-op demonstrations per task with a meta quest controller.

Algorithm Comparisons. We evaluated two prior methods for behavior cloning, the diffusion policy [3] as the image-based baseline, and 3D Diffusion policy (DP3) [4], which is recently proposed method that takes as inputs point-clouds. These methods demonstrate great performance in their respective input modalities. For our simulation tasks, we also evaluated an ablated version of method which we will refer to as GSFix. Instead of rendering from the foreground Gaussians, GSFix directly renders from all of the Gaussians. For both GSFix and GSMimic, we use diffusion policy with the only difference being inputs to the model.

Evaluation Protocol. For each task, we evaluated on 4 viewpoints of increasing difficulties: train view, close view (C), zoom out view (Z), and side view (S). In each view, we ensure that the objects of interest are still in sight. Please refer to the Appendix for a visualization of the views for each task. We reported success rate of each task evaluated at 100 and 10 different starting configurations for simulation and real-world tasks, respectively.

4.1.1 Experimental Results

We summarized our evaluation results for simulation tasks in Table 1 and real-world tasks Table 2.

3D Understanding of the Scene is Critical for View Robustness. As seen in the results, even though diffusion policy achieves great performance given observations from the training views, the success rate drops significantly even for the close view, a small perturbation to the training view, while the policy completely fails when the views are shifting farther away. The effect is even more drastic for more high-precision tasks like Tool Hang and Cup Unstacking (which requires the gripper to grasp on a thin edge). On the other hand, GSMimic achieves comparable performance at training views, while maintaining a reasonable performance across all testing views, demonstrating the importance of our dynamic 3D representation.

Learning with 2D Inputs Improves Task Performance. Similar to GSMimic, DP3 maintains a reasonable performance across different testing viewpoint. However, the task performance is in general considerably lower than the image-based models, especially for more complicated tasks. This highlights the current gap between learning directly from RGB inputs versus 3D representations, and the gap is likely to remain due to the abundance of 2D data and models. While on the other hand, our 3D representation has the flexibility to transform into 2D inputs, thus can better leverage rich semantics and achieve better task performance.

Rendering with Foreground Only is Crucial to Avoid Distribution Shift. If we directly render the Gaussians to obtain RGB inputs for training and testing as in GSFix, the task performance is still superior compared to diffusion policy (DP) at close views. However, at harder test views, the empty areas in the rendering due to limited field-of-view cause significant distribution shift, so that

Table 1: **Evaluation of Simulation Tasks Given Different Testing Viewpoints.** We present success rates of tasks with 100 different initial conditions under the train view and three test views: close view (C), zoom out view (Z), zoom out and side view (S).

	Lift				Can				Square				Tool Hang			
	Train	Test Views			Train	Test Views			Train	Test Views			Train	Test Views		
		C	Z	S		C	Z	S		C	Z	S		C	Z	S
DP	**0.98**	0.47	0.0	0.0	**0.93**	0.34	0.0	0.0	**0.82**	0.23	0.0	0.0	**0.64**	0.12	0.0	0.0
DP3	0.95	0.95	0.92	0.83	0.58	0.59	0.48	0.42	0.62	0.61	0.59	0.54	0.14	0.12	0.11	0.08
GSFix	**0.98**	0.85	0.80	0.07	0.91	0.87	0.67	0.03	0.80	0.23	0.00	0.00	0.60	0.15	0.00	0.0
GSMimic	**0.98**	**0.97**	**0.94**	**0.90**	0.92	**0.94**	**0.93**	**0.85**	0.81	**0.78**	**0.77**	**0.72**	0.62	**0.60**	**0.58**	**0.52**

Table 2: **Evaluation of Real-World Tasks Given Different Testing Viewpoints.** We present success rates of two real-world tasks with 10 different initial conditions, similarly from the training view and 3 test views.

	Stack Cups				Unstack Cups			
	train	close	zoom out	side	train	close	zoom out	side
DP	9/10	3/10	0/10	0/10	8/10	1/10	0/10	0/10
DP3	5/10	4/10	4/10	4/10	2/10	1/10	2/10	1/10
GSMimic	8/10	9/10	8/10	6/10	8/10	8/10	7/10	5/10

GSFix similarly fails. In fact, at harder testing views like side, occlusions still cause performance drops for GSMimic. This suggests possible augmentations to further handle distribution shifts in input observation for our future works.

4.2 Language-conditioned Dynamic Grasping

Evaluation Setup. We evaluated our method on language-conditioned dynamic grasping on two sets of five objects from a dining and a tool scene, as shown in Fig. 4. We first experiment on static grasping as a baseline. Then in the dynamic setting, we randomly move around the target objects when the robot is in action. For each object and setting, we repeats for 5 trials. As a baseline comparison, we remove object-centric updates, and directly optimize for the position of each Gaussian between updates (Object-Blind).

Evaluation Results. The results is presented in Table 3. From the results on static setting, we show that a semantic 3D representation is powerful, achieving a 86% success rate without demonstrations or other prior information. More importantly, our method still achieves a 72% success rate when objects are moving. This is only possible due to the dynamic aspect of our representation. We also show that our object-centric formulation is crucial, as the Object-Blind ablation completely fails to model object movements, making it impractical for dynamic scenes.

5 Related Work

Neural Dynamic Scene Representation. A pivotal advancement in neural volumetric scene representations was the introduction of Neural Radiance Fields (NeRF) [18], enabling high-quality renderings at novel views, which comes at the cost of prolonged training times. The recent development of 3D Gaussian Splatting (3D-GS) introduces a significant paradigm shift [1]. Unlike NeRF's implicit representation, 3D-GS utilizes explicit 3D Gaussian primitives, enabling scene representation, enabling fast, parallelizable rendering through rasterization. The explicit nature of 3D-GS, as opposed to the implicit form found in NeRF, has the potential for immediate updates in response to changes within the scene, making it particularly suited for dynamic environments. 3D-GS also led to several recent works that leverage the representation for offline dynamic scene reconstruction. The approaches include explicit parametrization of Gaussian parameters at different time steps and the modeling of a deformation field for Gaussians [19, 20, 21], which achieve high quality and fast rendering. These works highlight the potential for accurately capturing and rendering complex, dynamic scenes in real time. Nevertheless, they all require extensive viewpoints and offline training, while we aim at online updates with limited viewpoints for robotics applications.

Table 3: Evaluation of Language-conditioned Dynamic Grasping

	Dining					Tools					Total
	Green Bowl	White Bowl	Carrot	Snack	Spoon	Brush	Clamp	Screw driver	Tape	Mouse	
Static	5/5	5/5	5/5	4/5	4/5	5/5	3/5	4/5	4/5	4/5	43/50
Object-Blind	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	0/50
Ours	4/5	5/5	5/5	3/5	3/5	4/5	2/5	3/5	3/5	4/5	36/50

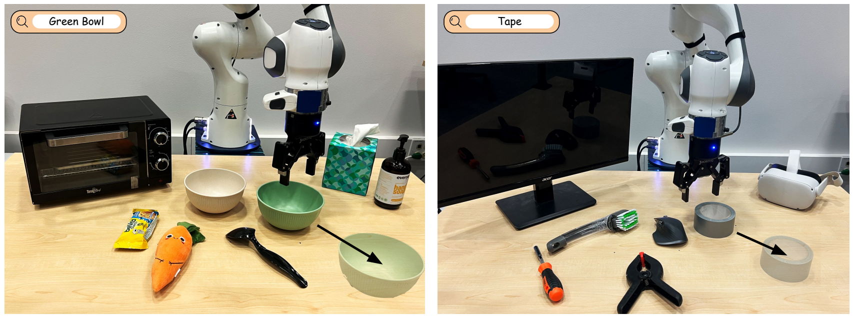


Figure 4: Language-conditioned Dynamic Grasping Task setup

3D Neural Representation for Robotic Manipulation. In the exploration of 3D representations for robotic manipulation, diverse approaches have leveraged neural fields [22, 23, 24, 25]. Among these, Neural Descriptor Fields stand out for constructing neural feature fields that generalize across different instances with minimal demonstrations, yet focus primarily on geometric rather than semantic features, limiting cross-category generalization [26]. Recent efforts have distilled neural feature fields using foundation models like CLIP [7] and DINO [6, 9] for supervision. Techniques such as F3RM [13] and LERF-TOGO [11, 12] have distilled neural feature fields to facilitate language-conditioned and task-oriented grasping, demonstrating the potential of foundation models in enhancing robotic manipulation. Despite these advancements, such methods often require dense camera views for training and retraining for new scenes, constraining their utility in dynamic settings. GNFactor attempts to address this by introducing a voxel encoder [27], yet the challenge of dense view dependency remains. Recently, D^{3}Fields proposed a dynamic and semantic 3D representation through 3D fusion, aiming for real-time updates with limited viewpoints [14]. However, D^{3}Fields requires feature extraction at every time step, increasing computational demands and complicating high-frequency reconstruction, highlighting a critical area for improvement in dynamic scene representation for robotic manipulation.

View-Generalization for Visuomotor Policies. In the field of robot learning, a primary challenge has been training models on limited views and achieving generalization to unseen views. Despite extensive efforts, such as those seen in the RoboNet [28] which amassed large-scale video datasets of various manipulation tasks, models pre-trained on these datasets still show poor performance, with success rates often below 20% on unseen camera viewpoints. Previous approaches to tackle this problem often extensive samples in simulation environments [29, 30], additional training viewpoints to create view-agnostic representations [31, 32, 33], or requires less scalable task-related inductive bias [34, 35]. Our simpler solution to the problem is to incorporate additional depth information and construct semantic and dynamic 3D representations allowing for effective projection back to training views, thus enhancing view generalization capabilities.

6 Discussion and Limitations

In this work, we propose to leverage 3D Gaussians as a semantic and dynamic 3D representation for robotics. We achieve a high update rate of 30 Hz with object-centric initialization and updates, which is sufficient for most robotic tasks. We demonstrate the practicality of our representation for training view-robust behavior cloning policies via GSMimic and language-conditioned dynamic grasping. However, a key limitation of our method is that in its current form, it does not introduce new Gaussians to represent possible new objects, which is crucial for extending the representation to open-world manipulation. We believe that with this extension, our proposed representation has the potential to apply to a wide range of in-the-wild robotic applications.

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)*, 42(4):1–14, 2023.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[3] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[4] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[5] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[6] 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*, pages 9650–9660, 2021.

[7] 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*, 139:8748–8763, 2021.

[8] 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.

[9] 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.

[10] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[11] 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.

[12] 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.

[13] 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.

[14] Y. Wang, Z. Li, M. Zhang, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li. D3 fields: Dynamic 3d descriptor fields for zero-shot generalizable robotic manipulation. *arXiv preprint arXiv:2309.16118*, 2023.

[15] 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.

[16] 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.

[17] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[19] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[20] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023.

[21] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

[22] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox. Rgb-d local implicit function for depth completion of transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4649–4658, 2021.

[23] Y. Wi, P. Florence, A. Zeng, and N. Fazeli. Virdo: Visio-tactile implicit representations of deformable objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3583–3590. IEEE, 2022.

[24] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg. Dexnerf: Using a neural radiance field to grasp transparent objects. In *5th Annual Conference on Robot Learning*, 2021.

[25] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Multi-task real robot learning with generalizable neural feature fields. In *Proceedings of the 7th Annual Conference on Robot Learning*, pages StartPage–EndPage, Location of the Conference, 2023. Publisher of the Proceedings, if available. Optional note, such as a DOI or a URL if the paper is available online.

[26] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[27] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[28] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[29] S. Yang, Y. Ze, and H. Xu. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] B. Chen, P. Abbeel, and D. Pathak. Unsupervised learning of visual 3d keypoints for control. In *International Conference on Machine Learning*, pages 1539–1549. PMLR, 2021.

[31] J. Shang and M. S. Ryoo. Self-supervised disentangled representation learning for third-person imitation learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 214–221. IEEE, 2021.

[32] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint. Reinforcement learning with neural radiance fields. *Advances in Neural Information Processing Systems*, 35:16931–16945, 2022.

[33] Y. Ze, N. Hansen, Y. Chen, M. Jain, and X. Wang. Visual reinforcement learning with self-supervised 3d representations. *IEEE Robotics and Automation Letters*, 8(5):2890–2897, 2023.

[34] P. Sharma, D. Pathak, and A. Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32, 2019.

[35] H.-Y. F. Tung, Z. Xian, M. Prabhudesai, S. Lal, and K. Fragkiadaki. 3d-oes: Viewpoint-invariant object-factorized environment simulators. *arXiv preprint arXiv:2011.06464*, 2020.

Supplementary: 3D-Aware Manipulation with Object-Centric Gaussian Splatting

Anonymous Author(s)
Affiliation
Address
email

1 Evaluation of Reconstruction Quality

Dataset and Metrics. Even though reconstruction quality is not the most important objective of our method, we present here some evaluation on the reconstruction quality. We make use of the data obtained through our teleoperated demonstrations. For all the data, we reconstruct the scenes with a training view and hold out an additional test view. For the metrics, we adopt the conventional reconstruction metrics: SSIM, PSNR, and LPIPS [1, 2]. To better present the metrics, we show the metrics at the initialization, and the percentage changes in the metrics in the following dynamic updates.

However, these are all global metrics that can be dominated by background reconstruction quality and thus overlook object movements in the dynamic scene, which is the main objective for robotic tasks. Thus, we also propose to use chamfer distance between the reconstructed foreground point-cloud $\mathcal{P}$ and the ground truth foreground point-cloud $\mathcal{P}_{gt}$.

$$CD(\mathcal{P}, \mathcal{P}_{gt}) = \sum_{x \in \mathcal{P}} \min_{y \in \mathcal{P}_{gt}} ||x - y||_2^2 + \sum_{y \in \mathcal{P}_{gt}} \min_{x \in \mathcal{P}} ||x - y||_2^2$$

We extract $\mathcal{P}$ by selecting the Gaussian centers x_i where $l_i \neq 0$. We run the full static Gaussian splatting algorithm, which takes much longer than our online reconstruction, to reconstruct the pseudo ground truth foreground point-cloud $\mathcal{P}_{gt}$.

Alternative Methods and Ablation. We compare our method with Dynamic 3D Gaussians (Dynamic-GS) [3], which directly optimizes the centers of each 3D Gaussian greedily. Even though the method is proposed for offline training, it is directly applicable to the online setting. We evaluate two variants of the method with different training steps per update, resulting in 1 Hz and 30 Hz update rates, respectively.

Necessity of Object-centric Updates. As shown in the evaluate results teleoperated dataset presented in Tab. 1, object-centric updates are crucial to represent robot arm and gripper movements in the scene. Without object-centric updates, with limited time budget, Dynamic-GS falls to a local minimum where the moving robot arm and object collapse to a single point. Only at 30x slower update rate, Dynamic-GS is able to faithfully reconstruct the movements.

Table 1: Quantitative Evaluation of Scenes from Teleoperated Demonstrations

	FPS	Last Frame				Average Frame			
		SSIM ↑	PSNR ↑	LPIPS ↓	CD ↓	SSIM ↑	PSNR ↑	LPIPS ↓	CD ↓
First Frame	-	0.8103	18.82	0.3528	0	0.8103	18.82	0.3528	0
Dynamic-GS (1Hz)	1	-6.87%	-9.51%	7.00%	0.008	-4.69%	-6.59%	4.42%	0.016
Dynamic-GS	30	-7.37%	-17.53%	16.50%	0.090	-4.66%	-11.96%	8.87%	0.045
Ours	30	-7.03%	-9.40%	8.99%	0.012	-4.12%	-5.53%	4.73%	0.017

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

2 Visualization of Evaluation Views for View-Robust Behavior Cloning

We visualize the evaluation viewpoints for the view-robust behavior cloning tasks in Fig. 1 below.

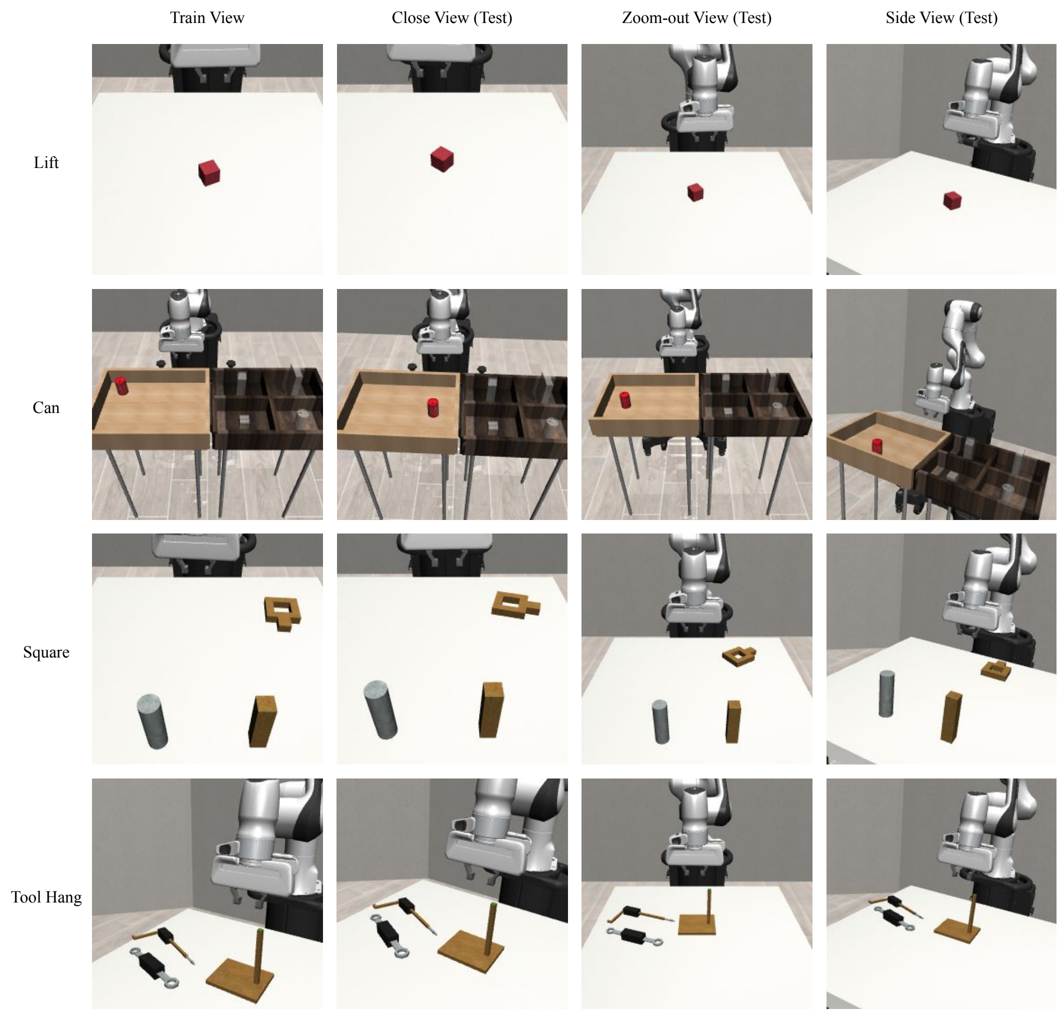


Figure 1: Evaluation views for view-robust behavior cloning.

References

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[2] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[3] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.