GAF: Gaussian Action Field as a Dynamic World Model for Robotic Manipulation

Ying Chai[*1], **Litao Deng**[*2], **Ruizhi Shao** [1], **Jiajun Zhang** [3], **Liangjun Xing** [1], **Hongwen Zhang** [2], **Yebin Liu** [1]
[1]Department of Automation, Tsinghua University
[2]School of Artificial Intelligence, Beijing Normal University
[3]School of Electronic Engineering, Beijing University of Posts and Telecommunications

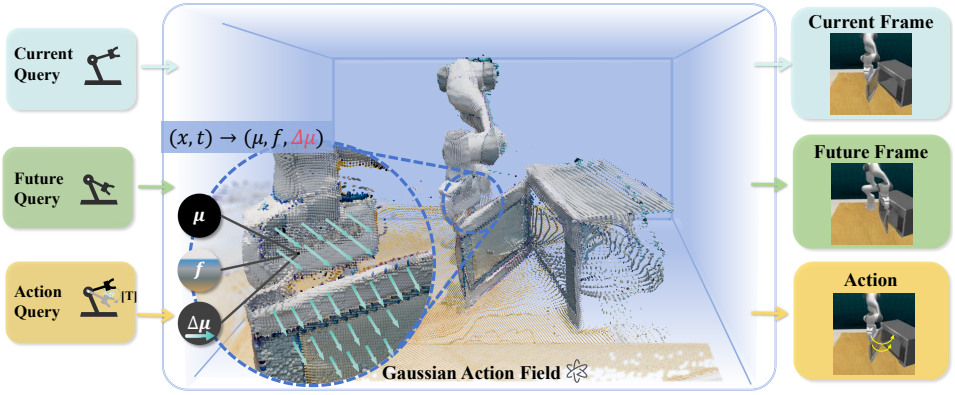

Figure 1: **Gaussian Action Field.** We present Gaussian Action Field (GAF), a dynamic world model where each 3D Gaussian is extended with motion attributes. This enables current scene rendering, future prediction, and action-aware motion learning, providing initial action hypotheses and serving as actionable guidance for robotic manipulation.

Abstract

Accurate action inference is critical for vision-based robotic manipulation. Existing approaches typically follow either a Vision-to-Action (***V-A***) paradigm, predicting actions directly from visual inputs, or a Vision-to-3D-to-Action (***V-3D-A***) paradigm, leveraging intermediate 3D representations. However, these methods often struggle with action inaccuracies due to the complexity and dynamic nature of manipulation scenes. In this paper, we propose a ***V-4D-A*** framework that enables direct action reasoning from motion-aware 4D representations via a Gaussian Action Field (GAF). GAF extends 3D Gaussian Splatting (3DGS) by incorporating learnable motion attributes, allowing simultaneous modeling of dynamic scenes and manipulation actions. To learn time-varying scene geometry and action-aware robot motion, GAF supports three key query types: reconstruction of the current scene, prediction of future frames, and estimation of initial action via robot motion. Furthermore, the high-quality current and future frames generated by GAF facilitate manipulation action refinement through a GAF-guided diffusion model. Extensive experiments demonstrate significant improvements, with GAF achieving +11.5385 dB PSNR and -0.5574 LPIPS improvements in reconstruction quality, while boosting the average success rate in robotic manipulation tasks by 10.33% over state-of-the-art methods.Project page: http://chaiying1.github.io/GAF.github.io/project_page/

[*]Equal contributions. Email: chaiyingchongchongchong@gmail.com, denglitao74@gmail.com.

Preprint. Under review.

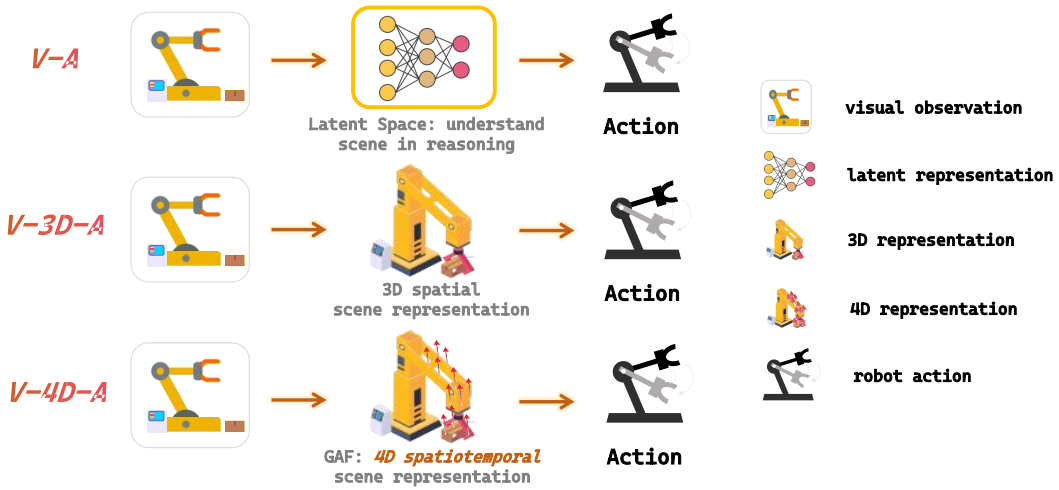

Figure 2: Comparisons between the previous ***V-A***, ***V-3D-A*** solutions and the proposed ***V-4D-A***.

1 Introduction

Effective perception is fundamental to robotic manipulation in unstructured 3D environments. Recent advances in vision-based methods [24, 38, 27, 66] have enabled robots to infer actions directly from visual observations by leveraging powerful foundation models [32, 58, 59, 11], which facilitates the high-level scene understanding and robotic manipulation.

Existing approaches for vision-based manipulation can be broadly categorized into two paradigms, as illustrated in Fig. 2. ***V-A*** (Vison-to-action) solutions [65, 53, 10, 2, 5, 33] directly map RGB observations to action sequences. While these methods benefit from end-to-end learning, they rely on implicit scene understanding and lack the modeling of 3D geometry, which is essential for fine-grained robotic manipulation. To address this issue, ***V-3D-A*** (vision-to-3D-to-action) solutions [15, 62, 61] incorporate 3D representations such as point clouds [13, 6, 64, 15, 29] and voxel grids [44, 26, 54, 41] to enable explicit geometric reasoning. Despite the geometric information provided by V-3D-A solutions, inferring action from 3D representations remains challenging, especially in cases of complex spatial structures and spatial relations. Besides, it is hard for the above two paradigms to model the temporal evolution of scene geometry, which introduces a disconnect between scene understanding and action generation in dynamic environments.

To take this into account, ManiGaussian [44] recently incorporated manipulation learning into a dynamic Gaussian Splatting framework. In their method, the action is first inferred via reinforcement learning and then used to deform the Gaussian for future scene consistency. From the aspect of action inference, ManiGaussian still belongs to the above two paradigms and suffers from the inaccurate perception of dynamic scene motion as shown in the experiments.

In this paper, our key motivation is that humans, when completing a task, envision how hands and objects might move and spatial relationships might change, and then act accordingly. Motivated by this, we aim to guide robotic action planning by explicitly modeling how scene geometry including robot changes over time. This process also aligns with the concept of a world model [16], where future scene dynamics are explicitly modeled to guide decision-making. To this end, we introduce a new paradigm, ***V-4D-A*** (vision-to-4D-to-action), which extends 3D representations with motion information to capture dynamic scene evolution, as shown in Fig. 2.

Unlike static representations that passively encode geometry, our 4D structure incorporates a latent world model that predicts the next-step scene evolution required to complete the task, based on the current observations. Such simultaneous dynamic perception and prediction enable more direct action inference since the scene motion inherently contains the movement trend information of the end-effector.

Specifically, we propose Gaussian Action Field (GAF) as a dynamic world model for robotic manipulation. To simultaneously model the dynamic scene and corresponding manipulation action,

GAF augments the 3DGS [31] representation with a learnable motion attribute that encodes the temporal displacement of each Gaussian, enabling the modeling of the dynamic scene and robotic geometry over time. This design enables three types of query functionalities within GAF, as shown in Fig. 1. The current query function supports view-consistent novel view synthesis of the present scene, facilitating accurate geometry understanding from two unposed RGB inputs. The future query function generates future scene states by applying motion attributes to the original Gaussians, providing supervision for learning temporal dynamics. The action query function supports the calculation of initial action by applying point cloud registration with the learned motion attributes.

Due to the noise in motion attributes, the initial action is typically inaccurate or ambiguous. To address this issue, we further introduce a diffusion-based neural refine module that predicts a refined and executable action. For more precise and temporally aligned robotic action generation, the denoising process is guided by the outputs of GAF, which acts as visual prompts with the motion attributes projected onto the current states for the predicted motion visualization. By modeling scene dynamics in a unified Gaussian world model, our V-4D-A paradigm enables coherent scene perception and robotic action, resulting in accurate, efficient, and temporally consistent manipulation.

GAF operates in a fully feed-forward manner and supports real-time execution on a single GPU during manipulation. Extensive experiments demonstrate that our method enables high-quality scene reconstruction, plausible future prediction, and accurate robotic manipulation, significantly outperforming V-A and V-3D-A baselines. Contributions of this work are summarized as follows:

- We propose a V-4D-A paradigm via Gaussian Action Field (GAF), which unifies the modeling of dynamic scene evolution and future-oriented action prediction, enabling more direct action reasoning from motion-aware 4D representations.
- We introduce three query types in GAF, namely current, future, and action, corresponding to different functionalities for spatial understanding, temporal prediction, and motion reasoning.
- We validate our method on robotic manipulation tasks, where it achieves state-of-the-art performance in both scene reconstruction quality and action generation.

2 Related Work

2.1 Vision-based Robot Learning

Vision plays a pivotal role in enabling robots to perceive and interact with their environments, and integrating visual perception into robotic manipulation tasks has been extensively studied [6, 2, 10, 13, 15, 7]. In such vision-based approaches, techniques like Vision-Language-Action (VLA), such as RT2 [4, 3], IGOR [8], ViLBERT [45], etc. [9, 1, 28, 35, 37, 36], have achieved impressive results by effectively combining visual information with language commands. In general, existing methods can be broadly categorized into 2D image-based approaches and 3D representation-based approaches. 2D methods typically rely on multi-view images as input, implicitly encoding 3D scene understanding within neural network reasoning. For example, GENIMA [53] utilizes Stable Diffusion [49] to generate images representing future robot poses, while SuSIE [2] and R&D [57] leverage diffusion models for sub-goal image generation and action refinement, respectively. These methods often struggle with accurately capturing precise 3D spatial relationships, limiting their effectiveness in high-precision tasks [6, 33]. In contrast, 3D representation-based methods, such as voxel grids and point clouds, explicitly model geometric structures, enabling more accurate spatial reasoning [56, 61, 15, 30]. For example, ManiGaussian [44] and GNFactor [61] both utilize voxel grids to represent the 3D scene; the former feeds these grids into a PerceiverIO [24]-based transformer policy using a reinforcement learning framework to get robot action, while the latter encodes them into a 3D semantic volumetric feature that is subsequently processed by a Perceiver Transformer [24] to predict actions. Additionally, Act3D [15] introduces a novel ghost point sampling mechanism to predict actions from semantic point clouds. These methods neglect the fact that, in addition to complex geometric structures and spatial relationships, robot learning also requires the consideration of time as a crucial dimension. Therefore, our 4D representation, which incorporates both spatial and temporal aspects, provides a more comprehensive task-level spatiotemporal representation, enabling better performance in robotics tasks.

2.2 World Model in Robotics

World models explicitly obtain environmental knowledge by constructing an internal representation that simulates the real world [14, 23, 16, 18, 19, 17, 20]. Through predicting the future states based on the current states, these methods successfully encode scene dynamics [22, 52]. Previous approaches utilize autoencoding to learn a latent space for predicting future states, achieving significant progress in simple tasks [16, 22, 50]. However, the limited representational ability of implicit features and the requirement for a large amount of data restrict their effectiveness and further applications. Recent approaches improve generalization by adopting explicit representations in image [12, 46] or language domains [40, 43, 66], leveraging rich semantics—e.g., UniPi [12] generates text-conditioned future frames, while Dynalang [40] predicts text-based states for navigation. However, they overlook the fact that the future is derived from the evolution of the current object's motion. These methods do not capture this process of motion. Instead, they predict future images [21], videos [39], point clouds [66], etc., and then infer what actions led to the changes in these visual representations. To model the motion in the scene, dynamic representations should be proposed as the internal representation of the world model for the scene. Among existing works, ManiGaussian [44] is the closest to this concept, which uses dynamic Gaussian point clouds as volumetric priors during training. However, this method does not use such dynamic representations during inference. They predict actions directly from static 3D representations. In contrast, our approach optimizing Gaussian Action Field for future state prediction during both training and inference phases. By eliminating dependencies on predefined modalities (e.g., pixels or text) and reducing data requirements, our method enables efficient and precise learning of scene-level dynamics, addressing key limitations of prior works.

3 Method

In this section, we introduce GAF, its implementation, and its application in robotic manipulation tasks. Sec. 3.1 defines the core representation and describes the three query modes. Sec. 3.2 details the technical implementation and overall network design. Sec. 3.3 illustrates how the outputs of GAF queries are used to generate executable actions.

3.1 Gaussian Action Field Representation

We define the Gaussian Action Field (GAF) as a unified spatiotemporal representation that associates each Gaussian primitive $\mathbf{g}(\mathbf{x})$ at time step t with both geometric attributes and motion dynamics. Formally, GAF is parameterized by a continuous function:

$$\mathcal{F}_\Theta : \{g(x), t\} \mapsto \{\mu, \Delta\mu, f\}\,, \tag{1}$$

where $\mu \in \mathbb{R}^3$ denotes the 3D position of the Gaussian, $\Delta\mu \in \mathbb{R}^3$ is the predicted displacement vector indicating temporal motion, and feature $f = \{c, \sigma, r, s\}$ represents the color, opacity, rotation, and scale attributes of each Gaussian.

The rendering process follows 3DGS [31], the pixel color at location $\mathbf{p}$ is computed using alpha-blend rendering:

$$C(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i c_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad \text{where, } \alpha_i = \sigma_i e^{-\frac{1}{2}(\mathbf{p}-\mu_i^{2d})^\top \Sigma_i^{-1}(\mathbf{p}-\mu_i^{2d})}, \tag{2}$$

where C is the rendered image, N denotes the number of Gaussians, α_i represents the 2D density of the Gaussian points in the splatting process, and Σ_i stands for the covariance matrix acquired from the rotation r and scales s.

To support current scene reconstruction, future state prediction, and action estimation, we define three types of queries over GAF, as defined in Eq. 3. The current query retrieves the position and feature parameters of Gaussians at the current time step, enabling rendering of the scene from novel views. The future query applies the predicted displacement to positions to obtain future positions, forming a temporally shifted Gaussian field for rendering future views. The action query retrieves motion attributes of manipulation-related Gaussians, and estimates the initial action via point cloud matching

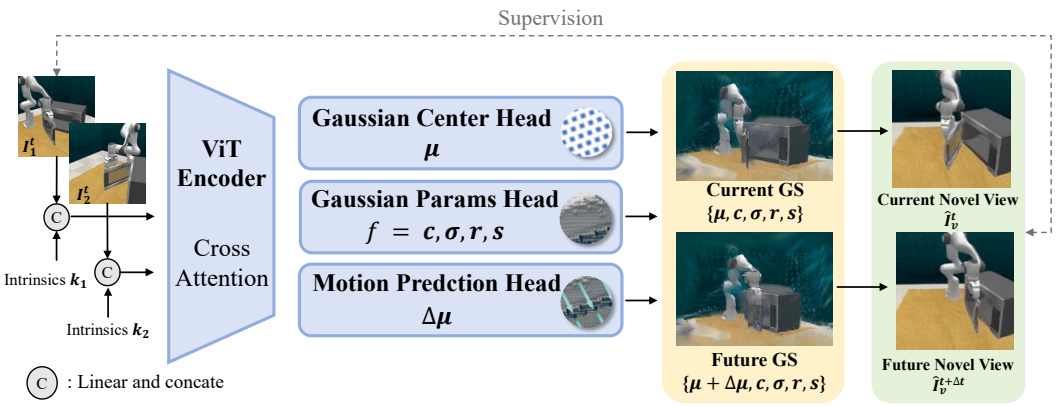

Figure 3: **Overview of GAF reconstruction.** Given sparse multi-view images, a Vision Transformer extracts hybrid scene features, which are decoded by three heads to predict Gaussian positions, motions, and appearance parameters, forming the GAF representation.

between current and future point clouds.

$$\begin{cases} \mathcal{Q}_{\text{current}}: \{g(x), t\} \xrightarrow[\mathcal{F}_\Theta]{\{\mu, f\}} GS_t \xrightarrow{\text{render}} I_t \\ \mathcal{Q}_{\text{future}}: \{g(x), t\} \xrightarrow[\mathcal{F}_\Theta]{\{\mu+\Delta\mu, f\}} GS_{t+\Delta t} \xrightarrow{\text{render}} I_{t+\Delta t} \\ \mathcal{Q}_{\text{action}}: \{g(x), t\} \xrightarrow[\mathcal{F}_\Theta]{\{\Delta\mu\}} A_{init} \end{cases} \tag{3}$$

3.2 Gaussian Action Field Architecture

The Gaussian Action Field (GAF) architecture unifies scene representation, dynamic motion prediction, and action reasoning. Our goal is to reconstruct motion-augmented Gaussians directly from sparse, unposed RGB inputs, enabling downstream temporal queries and manipulation control. Fig. 3 illustrates the overall design.

Dynamic Gaussian Reconstruction. GAF adopts a geometry-agnostic, pose-free approach for dynamic scene reconstruction, in contrast to traditional methods such as NeRF [47] and 3DGS [31], which rely on dense camera poses or strong geometric priors (e.g., cost volumes, epipolar constraints). Our architecture directly reconstructs high-fidelity motion-augmented Gaussians of input views in a canonical space aligned with the first input view. This is achieved using a feed-forward network that includes a vision transformer backbone and three specialized heads.

Specifically, given two unposed $H \times W$ images and their corresponding intrinsics $\{I_v^t, k_v^t\}_{v=1}^V$ at timestep t, we tokenize images into patch sequences and concatenate them. The resulting tokens are fed into a shared-weight Vision Transformer with cross-view attention to extract features.

For scene representation, we employ a decoupled two-head design $\mathcal{H}_{\text{Gauss}} = \{h_{\text{Center}}, h_{\text{Param}}\}$ based on the DPT architecture[48] to process the features: the Gaussian Center Head h_{Center} predicts only Gaussian centers, the Gaussian Param Head h_{Param} estimates the remaining parameters by additionally incorporating RGB information. The process can be formulated as:

$$\mathcal{H}_{\text{Gauss}}\left(\text{ViT}(\{\boldsymbol{I}_v^t, \boldsymbol{k}_v^t\})\right)_{v=1}^V = \{\mu_j^t, c_j^t, \sigma_j^t, r_j^t, s_j^t\}_{j=1}^{V\times H\times W}, \tag{4}$$

For scene dynamics, we introduce a Motion Prediction Head h_{Motion} following the same DPT-based architecture[48] as Gaussian Center Head. h_{Motion} predicts the per-point displacement $\Delta\mu_j^{t\to t+\Delta t}$, representing the motion of each Gaussian over a future interval Δt:

$$h_{\text{Motion}}\left(\text{ViT}(\{\boldsymbol{I}_v^t, \boldsymbol{k}_v^t\})\right)_{v=1}^V = \{\boldsymbol{\Delta\mu}_j^{t\to t+\Delta t}\}. \tag{5}$$

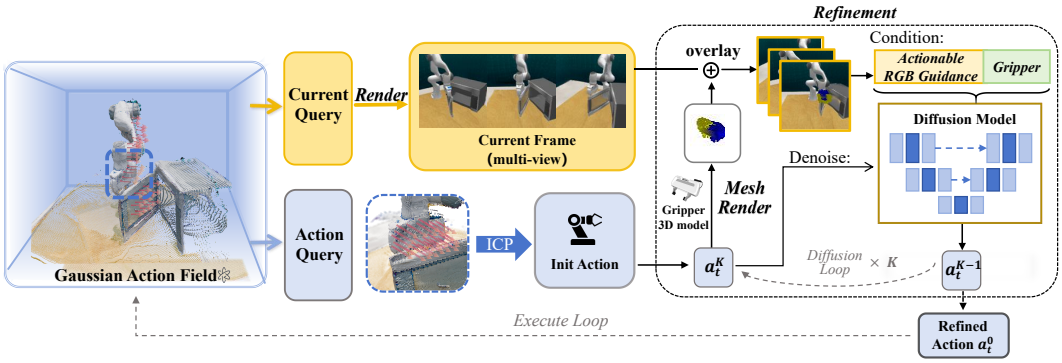

Figure 4: **Manipulation pipeline.** The GAF current and action queries provide current multi-view observations and an initial action estimate (left). These are then used as conditions for a refinement network to generate executable motion (right). The process repeats iteratively until the task completes.

The predicted displacement $\Delta\mu_j^{t\to t+\Delta t}$ are added to the current centers μ_j^t to obtain the future Gaussian positions $\mu_j^{t+\Delta t}$. These displaced centers are fused with the appearance and shape parameters $(c_j^t, \sigma_j^t, r_j^t, s_j^t)$ to form the future Gaussian field.

Deriving the current Gaussians and future Gaussians , we can render multiple novel view images for the current state $\{\hat{I}_v^t\}_{v=1}^M$ and future state $\{\hat{I}_v^{t+\Delta t}\}_{v=1}^M$, where M denote the number of synthesized views. This allows for direct RGB video frames supervision for the entire Dynamic Gaussian Reconstruction. The training process follows [60]:

$$\mathcal{L}_{\text{GAF}} = \mathcal{L}_{\text{LPIPS}}^t + \mathcal{L}_{\text{MSE}}^t + \mathcal{L}_{\text{LPIPS}}^{t+\Delta t} + \mathcal{L}_{\text{MSE}}^{t+\Delta t}. \tag{6}$$

where $\mathcal{L}^t$ enforces geometric fidelity to current observations and $\mathcal{L}^{t+\Delta t}$ regularizes future state prediction. They are aggregated into a unified objective, facilitating the joint optimization of motion-augmented Gaussians reconstruction.

Initial Action Computation. Given the reconstructed Gaussians at the current and future frames, we aim to explicitly describe the scene dynamics. Since our task focuses on robotic manipulation, we focus on the motion of the gripper, which serves as the robotic end-effector. Due to its rigid nature, we extract the manipulator-related Gaussians from the current state $\boldsymbol{\mu}_{\text{gripper}}^t$ and future state $\boldsymbol{\mu}_{\text{gripper}}^{t+\Delta t}$, and estimate a rigid transformation $T^{t\to t+\Delta t} \in \text{SE}(3)$ using ICP [51]. This transformation captures the gripper's motion and provides an explicit estimate of the scene dynamics:

$$T^{t\to t+\Delta t} = \arg\min \sum_{k\in gripper} \|T(\mu_k^t) - \mu_k^{t+\Delta t}\|^2 \tag{7}$$

$T^{t\to t+\Delta t}$ represents the change over Δt time steps. To obtain the transformation matrix for each time step during this period, we interpolate $T^{t\to t+\Delta t}$ to derive a sequence of transformation matrices. This sequence represents the initial action a_{init} that transitions the current frame to the future frame.

3.3 Manipulation with Gaussian Action Field

After introducing the definition and implementation of GAF, we now describe how it is deployed in robotic manipulation tasks. GAF supports three types of queries: While the visual outputs from current & future queries serve as supervision signals for GAF training, the initial actions obtained through action queries inevitably contain interaction-induced noise [42] due to partial observations, occlusions, or geometric ambiguities during physical interactions. Therefore, before executing these actions, we introduce a diffusion-based refinement module for action denoising. This module jointly leverages GAF-rendered multi-view observations (current query) and the initial action prediction (action query) to guide the diffusion model towards higher-quality denoising outcomes.

Diffusion-based Refinement. As illustrated in Fig 4, to fully leverage GAF's visual outputs and action predictions, we draw on insights from the R&D [57]. For each denoising step of duration

Δt , we project the initial action a_{init} corresponding gripper positions to pixel coordinates using camera parameters and then render gripper mesh onto current multi-view RGB images $\{\hat{I}_v^t\}_{v=1}^M$. This creates a unified representation, termed Actionable RGB Guidance, which integrates the visual 3D observations reconstructed by GAF with the temporally predicted actions. Such visual cues (surrounded by a yellow box in the refinement part on the right side of Fig. 4), along with initial action and gripper states, guide the diffusion model to minimize the following constraints:

$$\mathcal{L}_{refine} = L1(D, D^{gt}) + L1(\epsilon, \epsilon^{gt}) + BCE(g, g^{gt}) \tag{8}$$

where D represents denoising direction of gripper. ϵ is the noise added to the end-effector action. g is a binary variable that represents gripper's opening-closing action. $D^{gt}, \epsilon^{gt}, g^{gt}$ are their ground truth labels respectively. The denoised action sequence a_{refine} can be executed directly, enabling the acquisition of new observations of the updated scene. The entire pipeline, comprising GAF-based scene reconstruction, diffusion refinement and execution, is repeated iteratively until the manipulation task is completed. This closed-loop framework enables continuous adaptation to dynamic scene changes, leveraging GAF's spatiotemporal reasoning to maintain robust performance under occlusion and interaction uncertainties.

4 Experiments

In this section, we first introduce the setup of the experiment including data and baseline methods. Then, to thoroughly assess the effectiveness of Gaussian Action Field in scene representation, future state prediction, and accurate action prediction, we evaluate our framework in dynamic scene reconstruction and task-level success rate. Finally, we conduct an ablation study to further validate the effectiveness of the various components within our model.

Simulation. For manipulation tasks, we select 9 tasks from popular RLBench[25] tasks, covering diverse manipulation challenges including articulated object handling and occlusion-rich interactions. To ensure generalization, we randomly initialize the objects in the environment and collect 20 demonstrations for training phase of each individual task. To eliminate randomness and ensure representational generalization, we conduct evaluations across 100 episodes for each task, and the objects are also initialized randomly. For visual data collection, we collect 30 views RGB sequences using a circular camera array centered on the robot workspace like GNFactor[61].

Baselines. For scene reconstruction quality, we compare against ManiGaussian[44] which also involves the reconstruction of current and future Gaussians during the training process.

For task success rate, ACT [65] and DP [10] are two classic methods in robotics, while the R&D [57] sub-method R&D-AI, which integrates actions and images, represents the state of the art (SOTA) in RLBench tasks. All three methods belong to the ***V-A*** category. ManiGaussian is classified under the ***V-3D-A*** category for understanding the sceneand then predicting actions from 3D representations. To ensure fairness, all baselines adopt identical camera configurations, action spaces, and task variations. Parameter settings are detailed in the supplementary material.

4.1 Evaluation on Scene Reconstruction and Prediction

To validate reconstruction capabilities of our Gaussian Action Field, we compare its with ManiGaussian. Although Manigaussian uses static 3D Gaussian representations, it applies deformation to the current Gaussian and obtain the future Gaussian to evaluate action quality. As a result, both methods generate Gaussian point clouds for current and future frame during training.

Qualitative Analysis. As illustrated in Figure 5, our method achieves superior reconstruction fidelity and novel-view synthesis. ManiGaussian's renders (up) exhibit blurred textures and incomplete geometric details resulting in ambiguous spatial relationships. In contrast, our renders (down) preserve fine geometric structures, such as the gripper's articulated joints and object surfaces, even under partial observations. This clarity in reconstructing the Gaussian point cloud allows for the extraction of precise end-effector point clouds to calculate the action. This contributes to the fundamental difference compared to Manigaussian.

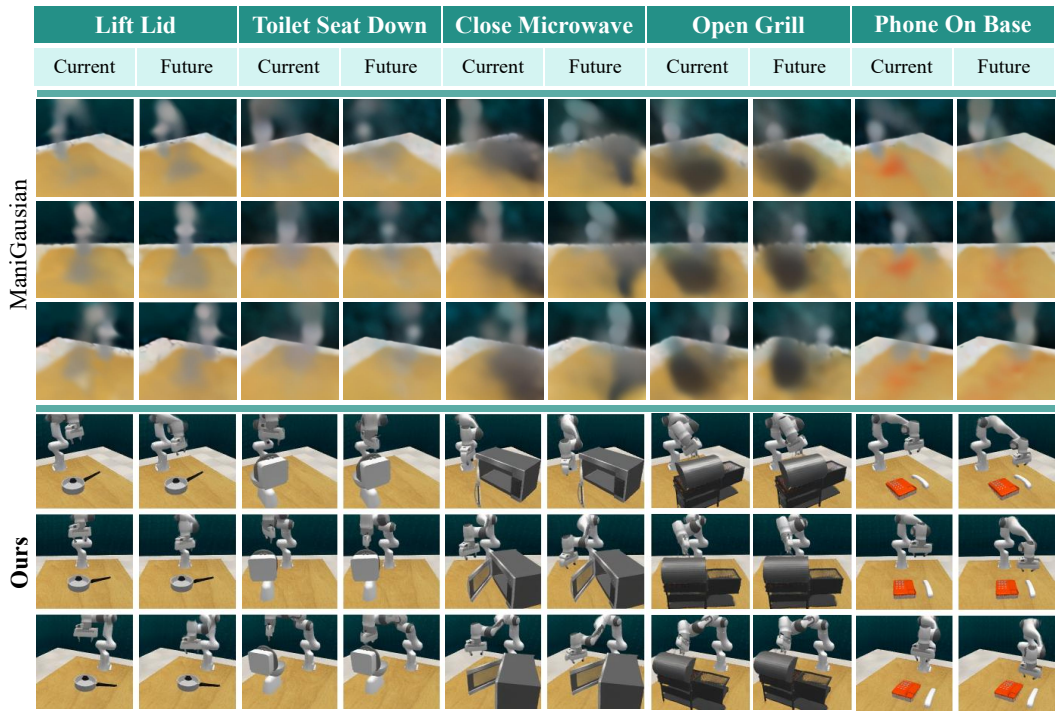

Figure 5: Comparison of current scene reconstruction and future scene prediction from novel views.

Quantitative Metrics. We further evaluate reconstruction quality using standard metrics: PSNR (photometric fidelity), SSIM (structural similarity), and LPIPS (perceptual consistency). As shown in Table 1, our method outperforms ManiGaussian by +11.5385 dB PSNR, +0.3864 SSIM, and -0.5574 LPIPS on average across tasks in current scene reconstruction, and +10.5311 dB PSNR, +0.3856 SSIM, and -0.5757 LPIPS in future state prediction. These metrics confirm that our dynamic rendering framework ensures high quality geometric accuracy and temporal coherence.

Table 1: Current & Future Novel view synthesis performance Comparison.

Method	Close Microwave			Toilet Seat Down			Lift Lid			**Average**		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ManiGaussian[44] /Now	16.4274	0.3753	0.7628	16.5628	0.3976	0.6806	16.1492	0.4139	0.6217	16.3798	0.3956	0.6884
ManiGaussian[44] /Future	16.1368	0.3565	0.7896	15.7953	0.3687	0.7161	15.3727	0.3969	0.6572	15.7683	0.3740	0.7210
Ours / Now	27.0986	0.7976	0.1291	28.1652	0.7779	0.1352	28.4912	0.7705	0.1286	**27.9183**	**0.7820**	**0.1310**
Ours / Future	24.5881	0.7650	0.1489	27.2951	0.7655	0.1456	27.0150	0.7483	0.1413	**26.2994**	**0.7596**	**0.1453**

↑: Higher is better; ↓: Lower is better.

4.2 Evaluation on Manipulation Success Rate

We compare our GAF with baselines on success rates across 9 RLBench tasks, focusing on precision manipulation, occluded interactions, and dynamic contact to investigate how our GAF, a ***V-4D-A*** method for scene evolution and action prediction, contributes to the improvement in action-level prediction accuracy.

Result and Discussion. Quantitative results are presented in Table 3. As indicated by the results, our approach, which explicitly models 4D scene variations, outperforms baseline ***V-A*** methods ACT, DP and R&D that understand the scene in latent space. Compared with current SOTA R&D-AI, our model achieves a 18% improvement in the task "Toilet Seat Down" and a 14% improvement in the task "Close Laptop". In the 'Toilet Seat Down' task, the robot needs to accurately perceive the

Table 2: Success rates (%) of the baselines and ours variant evaluated on RLBench tasks.

Method	Toilet Seat Down	Open Grill	Close Grill	Close Microwave	Close Fridge	LIFT LID	Phone On Base	Lamp On	Close Laptop	Average
ACT [65]	46	15	33	59	18	89	25	20	61	40.67
DP [10]	39	20	36	62	16	81	23	21	64	40.22
R&D-AI [57]	53	17	38	70	22	91	25	**23**	55	43.78
ManiGaussian [44]	31	19	28	67	26	87	18	20	57	39.22
Ours	**71**	**26**	**55**	**83**	**34**	**100**	**28**	21	**69**	**54.11**

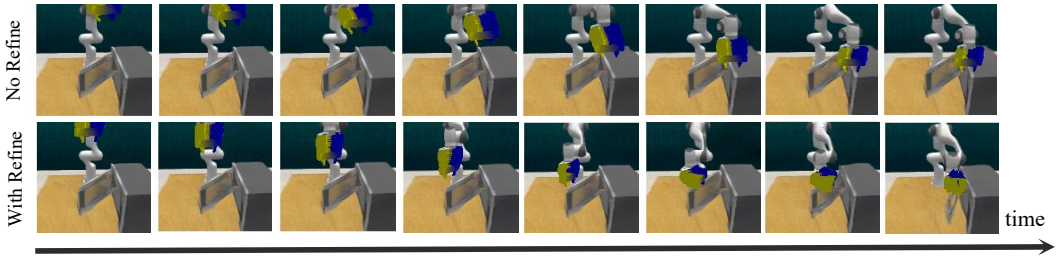

Figure 6: **Ablation action** The upper image shows a failed experiment without action refinement, while the lower image depicts a successful experiment after action refinement.

orientation and position of the seat in relation to the surrounding environment, such as the toilet and the seat. A 3D representation of the scene allows the robot to model these spatial relationships. This demonstrates how 3D scene modeling is fundamental to improving a robot's ability to understand its environment and execute tasks with higher accuracy.

To further validate the effectiveness of spatiotemporal representation, we compare our model with ***V-3D-A*** method Manigaussian. Our approach exhibits higher spatiotemporal scene understanding (specifically, reconstruction quality in Section 4.1), resulting in improved success rates across nearly all tasks. This is because our 4D representation includes dynamics modeling along the temporal dimension, allowing for more accurate task execution when inferring actions.

4.3 Ablation Study

Ablation on Gaussian Action Field. To evaluate the contribution of the Gaussian Action Field, we remove this component and directly predict actions from two images by using a diffusion model without multi-view rendering or initial action priors. This setup aligns with generative baselines like DP and R&D. As shown in Table 3, our full method outperforms R&D-AI by +10.33% in average success rate across tasks. Notably, in occlusion-heavy tasks like "Close Microwave," our method achieves a 13% improvement over R&D-AI, demonstrating the critical role of explicit scene reconstruction in resolving spatial ambiguities. These results demonstrate that modeling 3D geometry and dynamics through motion fields significantly enhances action prediction robustness.

Ablation on Action Refinement. We analyze the effectiveness of action refinement by comparing the initial action (directly derived from Gaussian motions) and the refined action. From the comparison in the Fig. 6, it can be observed that before contact, the initial action often aligns well with the target object's pose. However, during interaction, reconstruction errors from partial occlusions observations lead to physically implausible robot object relations (e.g., misaligned contacts or penetration), requiring action refinement. For example, in the shown experiment "Close Microwave", the initial action directly instructs robot to move toward the closing area without considering that the object's geometric appearance should be manipulated from the door. The diffusion-based refinement corrects these errors by supervised learning using real physical interactive data.

5 Discussion

We present Gaussian Action Field (GAF), a V-4D-A paradigm that infers future evolution of a scene from current visual observations to guide robotic manipulation. GAF supports scene reconstruction, future prediction, and action generation within a unified Gaussian world model. This feed-forward pipeline requires only two unposed RGB images, operates without any heavy setup, and supports real-time execution. Experiments on RLBench demonstrate that GAF achieves superior performance in both reconstruction and manipulation tasks. While our current method focuses on geometric modeling and motion prediction, it lacks semantic or task-level understanding. Future work will incorporate language modeling to bring high-level semantic priors into the system, extending our framework toward VL-4D-A (Vision-Language-4D-to-Action) to support context-aware manipulation.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[2] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *ArXiv*, abs/2310.10639, 2023.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[5] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *ArXiv*, abs/1912.08830, 2019.

[6] Shizhe Chen, Ricardo Garcia Pinel, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *ArXiv*, abs/2309.15596, 2023.

[7] Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiangyu Chen, Min Xu, Ruizhen Hu, Hongyuan Zhang, Xuelong Li, and Ping Luo. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. *ArXiv*, abs/2411.18369, 2024.

[8] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2020.

[10] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[12] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[13] Chongkai Gao, Zhengrong Xue, Shuying Deng, Tianhai Liang, Siqi Yang, Lin Shao, and Huazhe Xu. Riemann: Near real-time se (3)-equivariant robot manipulation without point cloud segmentation. *arXiv preprint arXiv:2403.19460*, 2024.

[14] Zeyu Gao, Yao Mu, Chen Chen, Jingliang Duan, Ping Luo, Yanfeng Lu, and Shengbo Eben Li. Enhance sample efficiency and robustness of end-to-end urban autonomous driving via semantic masked world model. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[15] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 1(3), 2023.

[16] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

[17] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

[18] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[19] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[20] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[21] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[22] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[23] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022.

[24] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. *ArXiv*, abs/2103.03206, 2021.

[25] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[26] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13729–13738, 2021.

[27] Mazeyu Ji, Ri-Zhao Qiu, Xueyan Zou, and Xiaolong Wang. Graspsplats: Efficient manipulation with 3d feature splatting. *arXiv preprint arXiv:2409.02084*, 2024.

[28] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[29] Jian-Jian Jiang, Xiao-Ming Wu, Yi-Xiang He, Ling an Zeng, Yi-Lin Wei, Dandan Zhang, and Wei-Shi Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework. *ArXiv*, abs/2503.09186, 2025.

[30] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *ArXiv*, abs/2402.10885, 2024.

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[33] Alina Kloss, Maria Bauza, Jiajun Wu, Joshua B Tenenbaum, Alberto Rodriguez, and Jeannette Bohg. Accurate vision-based manipulation through contact reasoning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6738–6744. IEEE, 2020.

[34] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[35] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11336–11344, 2020.

[36] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[37] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[38] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

[39] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *ArXiv*, abs/2406.16862, 2024.

[40] Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. *arXiv preprint arXiv:2308.01399*, 2023.

[41] I-Chun Arthur Liu, Sicheng He, Daniel Seita, and Gaurav Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. In *Conference on Robot Learning*, 2024.

[42] Xueyi Liu and Li Yi. Geneoh diffusion: Towards generalizable hand-object interaction denoising via denoising diffusion, 2024.

[43] Guanxing Lu, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Thinkbot: Embodied instruction following with thought chain reasoning. *arXiv preprint arXiv:2312.07062*, 2023.

[44] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. *arXiv preprint arXiv:2403.08321*, 2024.

[45] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[46] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023.

[47] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[48] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[49] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[50] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[51] Aleksandr V. Segal, Dirk Hähnel, and Sebastian Thrun. Generalized-icp. In *Robotics: Science and Systems*, 2009.

[52] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.

[53] Mohit Shridhar, Yat Long Lo, and Stephen James. Generative image as action models. *arXiv preprint arXiv:2407.07875*, 2024.

[54] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *ArXiv*, abs/2209.05451, 2022.

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[56] Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion. *ArXiv*, abs/2411.12633, 2024.

[57] Vitalis Vosylius, Younggyo Seo, Jafar Uruç, and Stephen James. Render and diffuse: Aligning image and action spaces for diffusion-based behaviour cloning. *arXiv preprint arXiv:2405.18196*, 2024.

[58] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[59] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[60] Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024.

[61] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[62] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations, 2024.

[63] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[64] Tong Zhang, Yingdong Hu, Hanchen Cui, Hang Zhao, and Yang Gao. A universal semantic-geometric representation for robotic manipulation. *arXiv preprint arXiv:2306.10474*, 2023.

[65] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[66] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *ArXiv*, abs/2403.09631, 2024.

A Additional Experiments

In this section, we designed additional experiments to demonstrate the performance of GAF. We mainly evaluate its ability in spatial generalization, data efficiency, and multi-task learning.

A.1 Spatial Generalization

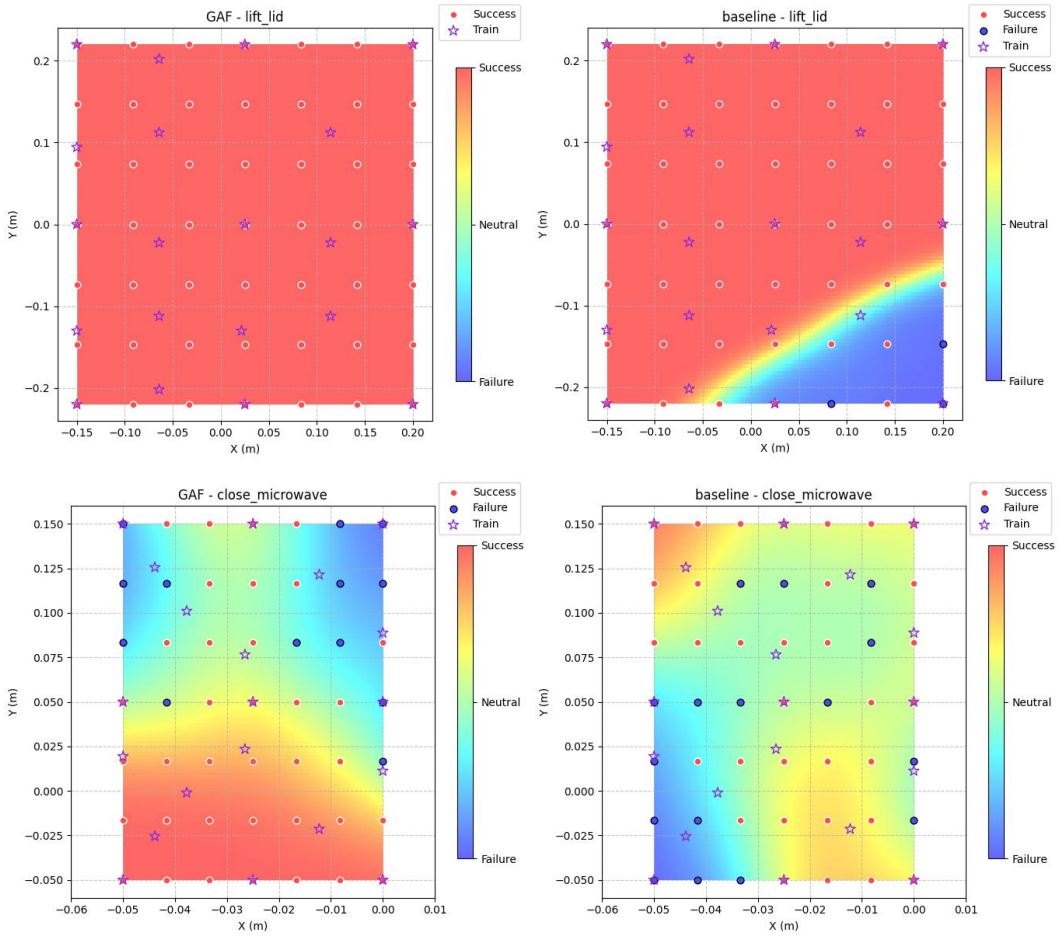

Figure 7: **Spatial Generalization.** Outcome of GAF and baseline trained on 20 demonstrations (purple stars). The heat maps represent Gaussian kernel density estimations for relative likelihood polarity over the workspace, with red and blue colours representing successes and failures, respectively.

We propose a systematic data collection strategy to ensure comprehensive spatial coverage of object poses within the operational workspace. The methodology initiates with a canonical demonstration where the object is positioned at the workspace centroid (x_0, y_0). Subsequently, we implement an

iterative farthest-point sampling algorithm that selects subsequent poses by maximizing the minimum Euclidean distance to existing samples in the demonstration set $\mathcal{D} = \{p_i\}_{i=1}^{n}$, formally expressed as:

$$p_{n+1} = \arg\max_{p \in \mathcal{P}} \min_{p' \in \mathcal{D}} ||p - p'||_2$$

where $\mathcal{P}$ denotes the feasible pose space and $||\cdot||_2$ represents the L2-norm. During evaluation, we establish a systematic protocol employing a dense grid sampling methodology across the entire workspace. This experimental design guarantees sufficient spatial variation in test conditions while maintaining measurement consistency, with comparative analysis performed against the baseline R&D [57] in Figure 7.

As illustrated in Figure 7, R&D encounters significant challenges when objects are placed along the boundaries and corners of the workspace. Notably, in the close_microwave manipulation task, previous methods exhibits pronounced performance degradation even within central operational regions. In contrast, our method achieves superior spatial generalization capability even when objects are placed on boundaries. Besides, our method is less sensitive to corner areas.

A.2 Data Efficiency

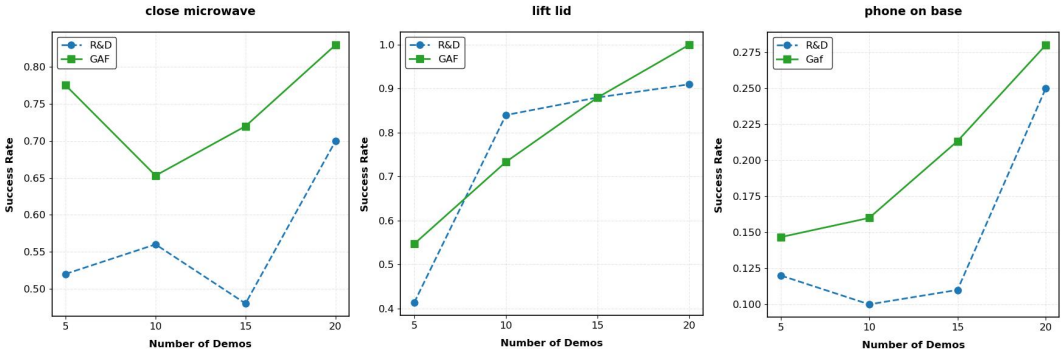

Figure 8: **Data Efficiency.** The success rate of our method and the baseline R&D in three tasks (Lift Lid, Close Microwave, Phone On Base) varies with different demonstrations.

For this set of experiments we train the models on different numbers of demonstrations collected in the same data collection strategy as in A.1 and evaluate them in a grid-like manner to ensure that the experiments present a sufficient level of challenge. Figure 8 shows how the performance of our GAF and current SOTA R&D change with increasing density of the workspace coverage, i.e., number of demos. As expected, all the methods benefit from larger amounts of demonstrations. Moreover, GAF achieves 90% of the peak performance with 15 demos, demonstrating excellent data efficiency.

A.3 Multi-task Test

Table 3: Performance of GAF and baseline, when training a single model using demonstrations from 4 different tasks (20 demonstrations per task), we show the success rate (%) and the performance difference compared to the single-task setting.

	Toilet Seat Down	Close Microwave	LIFT LID	Close Laptop	**Average**
R&D [57]	65 (+12)	50 (-20)	22.5 (-68.5)	17.5 (-37.5)	38.75 (-28.5)
Ours	**59 (-12)**	**85 (+2)**	**57 (-43)**	**79 (+10)**	**70 (-10.7)**

In our previous experiments, we trained distinct policy networks for each individual task. To validate the generalization capabilities of GAF, we test its capacity to learn multiple tasks simultaneously. It is a critical property for any world model.

In this section, we train a single network using data collected from 4 RLBench tasks, 20 demonstrations each. Object positions are randomly initialized in both data collection and model evaluation.

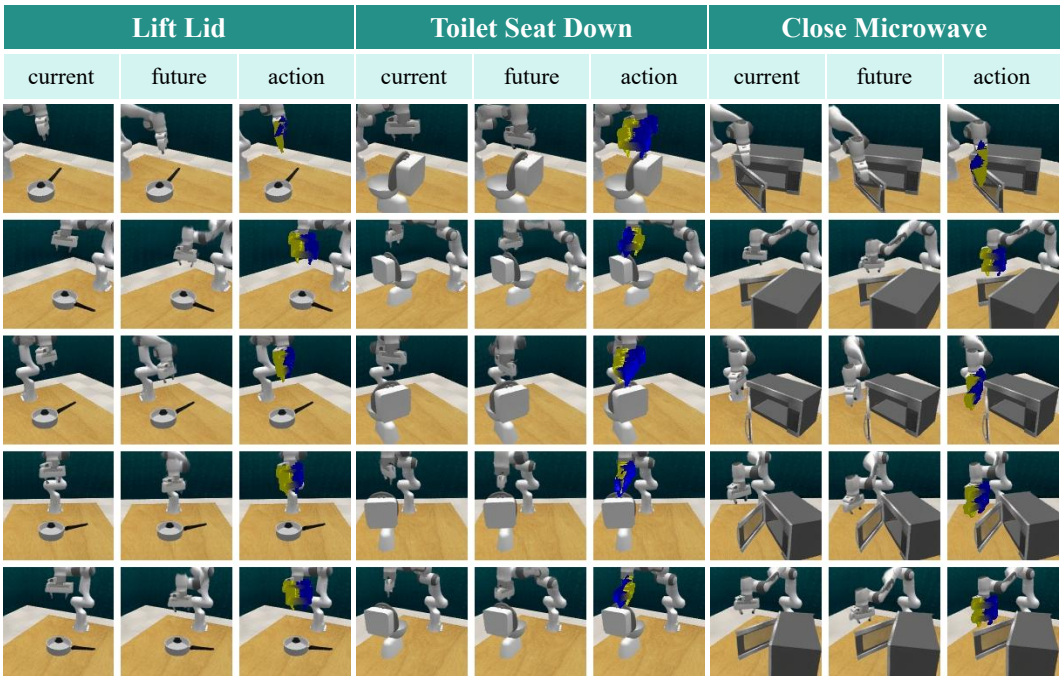

Figure 9: **GAF Query Result.** Multiview images rendered from current and future Gaussian point clouds, along with the predicted initial actions visulization.

As table 3 illustrated, our method's average success rate only declines 10.7%. This highlights GAF's robust multi-tasking capabilities, underscoring its effectiveness as a world model-based approach. Our success rate exhibits the most significant decline in the "lift lid" task, which is markedly distinct from the other three tasks. Nevertheless, in comparison to the substantial 28.5% decline observed in the baseline, our method demonstrates considerably superior performance.

B GAF Query Result

Figure 9 presents the results obtained by querying GAF with the current query, future query, and action query. The "current" column, "future" column, and "action" column represent, respectively, the multiview images rendered from the current Gaussian point cloud reconstructed by GAF, the multiview images rendered from the predicted future Gaussian point cloud, and the initial action computed based on these two Gaussian point clouds (visualized by rendering the mesh into the images). We show results on three tasks from RLBench: "lift lid," "toilet seat down," and "close microwave." It can be observed that our method produces clear novel views RGB for both the current and future states, and is capable of generating reasonable initial actions based on the transformation from the current to the future state.

C Implement Details

Training Phase During the training of GAF, The network is end-to-end trained using ground truth target RGB video frames as supervision with a linear combination of MSE and LPIPS [63] loss with weights of 1 and 0.05, respectively. We initialize the ViT, Gaussian center head and motion prediction head with the weights from MASt3R [34], while the remaining layers are initialized randomly. The GAF model is trained on 9 separate tasks, each consisting of 20 demonstrations and 200 input RGB image video frames per demonstration. It have been trained for 80k iterations (with a batch size of 16) The model is trained using a single NVIDIA RTX A800 GPU, which takes approximately 24 hours to complete.

In the action refinement process, we use 50 diffusion ierations based on DDIM [55]. To obtain more precise local observations, we incorporated the GT wrist camera data as an auxiliary resource in this

section. We use 2 last observations as input and predict 8 future actions. It have been trained for 50k iterations (with a batch size of 8). The denoising process completes in 1.5 days on a single NVIDIA RTX A4090 GPU without extensive optimisation.

Evaluation Phase For a fair comparison, all methods, including the baselines, utilize RGB observations (128×128) from two external cameras and another wrist camera. Different from training parameters, during the inference phase, we only need 3 diffusion iterations, which makes our online deployment more real-time. The hyperparameters used in GAF are shown in Table 4. Other baselines hyperparameters are in line with previous works [57, 65, 44, 10] for fair comparison.

Table 4: Hyperparameters

Hyperparameter	Value
GAF training iteration	80k
Refine training iteration	50k
image resolution	128×128
optimizer	AdamW
GAF weight decay	0.05
Refinement weight decay	0.01
GAF learning rate	2e-5
Refinement learning rate	1e-4
Number of Gaussian points	131072

D RLBench Dataset Success Metric

In this section, we provide a precise overview of the RLBench [25] dataset. We describe each of the 9 tasks in detail, including key action and success metrics.

D.1 Toilet Seat Down

Description: The robot must lower the toilet seat from an upright position to a closed position.

Key Actions: Grasp the toilet seat. Apply a controlled downward motion to close it.

Success Metric:The toilet seat is fully lowered, resting flat on the toilet bowl, and remains stationary.

D.2 Open Grill

Description: The robot needs to open the lid of a grill (e.g., a barbecue grill).

Key Actions: Grasp the grill lid handle. Pull and lift the handle to open the lid.

Success Metric: The grill lid is fully open and remains stationary in the open position.

D.3 Close Grill

Description: The robot must close the lid of the grill after it has been opened.

Key Actions: Grasp the grill lid handle. Push and twist the lid down to close it.

Success Metric: The grill lid is fully closed, flush with the grill body, and does not rebound.

D.4 Close Microwave

Description: The robot must close the door of a microwave that has been left open.

Key Actions: Push the microwave door. Apply force to swing the door shut.

Success Metric:The microwave door is fully closed.

D.5 Close Fridge

Description: The robot needs to close the door of a refrigerator.

Key Actions: Grasp or push the fridge door. Apply force to close the door completely.

Success Metric:The fridge door is fully closed.

D.6 LIFT LID

Description: The robot must lift the lid of a saucepan.

Key Actions: Grasp the lid handle. Lift the lid upward and away from the saucepan.

Success Metric:The lid is completely removed from the container and held in a stable position without contact with the container.

D.7 Phone On Base

Description: The robot must place a phone back onto its base.

Key Actions: Grasp the phone. Align it with the base. Place it gently onto the base.

Success Metric: The phone is securely placed on the base, properly aligned.

D.8 Lamp On

Description: The robot must turn on a lamp, typically by interacting with a button.

Key Actions: Locate the lamp's activation mechanism. Interact with the mechanism to turn the lamp on.

Success Metric:The lamp emits light, indicating it has been successfully turned on.

D.9 Close Laptop

Description: The robot must close the lid of an open laptop.

Key Actions: Grasp the laptop lid. Push and twist the lid down to close it.

Success Metric:The laptop lid is fully closed, with no visible gap between the lid and the base.