Vista3D: Scene-Aware Vision-Language-Action Model using 3D Gaussian Splatting

Abhishek Mathur*, Ishita Gupta*, Rodrigo Lopes Catto*, and Tom Gao*
Carnegie Mellon University, Pittsburgh, USA
{armathur, ishitag, rlopesca, zimingg}@andrew.cmu.edu
* Equal Contribution

***Abstract*—We propose Vista3D, a 3D Gaussian Vision-Language-Action (VLA) model that augments existing VLA architectures (such as NVIDIA GR00T N1.5 [1]) with an explicit 3D vision head capable of processing Gaussian Splat scene representations. While current VLA systems demonstrate strong generalization, their dependence on purely 2D visual inputs restricts their ability to perform tasks requiring depth awareness, occlusion reasoning, and precise spatial grounding. Vista3D addresses this limitation by integrating a semantically enriched Gaussian Splat map and a learned querying module that enables explicit 3D goal localization, improving robustness in real-world manipulation settings.**

***Index Terms*—3D Gaussian Splatting, Vision-Language-Action Models, GR00T, Robotic Manipulation, 3D Perception, Real World Tasks**

I. Introduction

Recent Vision-Language-Action (VLA) models such as GR00T have shown strong generalization in robotic manipulation tasks. These models utilize a pretrained VLM backbone to convert language and visual instructions into tokens that the downstream model may decode into robot action. Yet in practice, these models rely heavily on 2D inputs, which limits their ability to reason about depth, occlusion, and geometry—key elements for real-world manipulation.

At the same time, three-dimensional scene representation methods such as Gaussian Splatting and Neural Radiance Fields (NeRF) have advanced significantly in capturing detailed and continuous 3D geometry and appearance. Integrating these representations into VLM and VLA frameworks can improve spatial understanding and overall manipulation robustness.

This project aims to extend GR00T with a 3D vision head capable of processing Gaussian Splat inputs. Using datasets collected from a Kinova robotic arm, we will fine-tune and benchmark the model to evaluate how 3D perception improves visuomotor performance compared to standard 2D vision baselines.

II. Related Work

A. Vision-Language-Action models

Vision-Language-Action (VLA) models such as PaLM-E, RT-2, and GR00T [1]–[3] combine visual perception and language understanding to perform robotic manipulation tasks. These models show strong generalization to unseen instructions and environments but still rely mainly on 2D visual inputs, which limits their spatial understanding. More recent works, including SmolVLA and HybridVLA [4], [5], focus on improving efficiency and long-horizon planning, yet they also depend on 2D image representations.

B. 3D Scene Representations

3D scene representation methods aim to capture the geometry and structure of real-world environments. Neural Radiance Fields (NeRF) introduced continuous 3D reconstruction from images but are computationally expensive. 3D Gaussian Splatting [11] improved reconstruction efficiency by representing scenes with Gaussian primitives that allow real-time rendering. Later works such as LEGaussians and SplatMover [6], [7] applied these ideas to robotic domains, making 3D reconstruction faster and more useful for manipulation tasks.

Recent approaches like 3D Diffusion Policy and Improved 3D Diffusion Policy (iDP3) [4], [5] use point clouds for 3D control learning, while NVIDIA MindMap [10] employs NVBlox-based

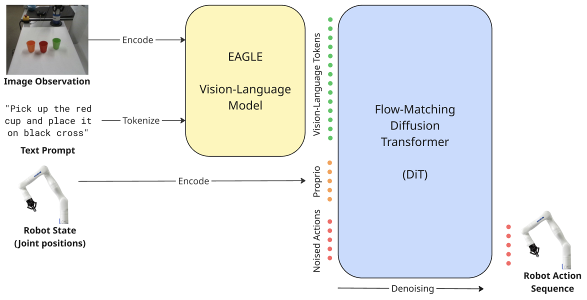

Fig. 1. The original GR00T architecture, applied to our Kinova manipulator system.

voxel maps for spatial memory. MolmoAct [13] extends large multimodal models for embodied reasoning using multi-view image features and learned depth priors to approximate 3D structure, but it still lacks an explicit geometric representation. However, both point-cloud and voxel-grid methods often lose surface continuity and fine texture detail. Gaussian Splatting, in contrast, provides a smooth, continuous, and compact 3D representation with efficient storage and differentiable rendering, making it more suitable for integration into multimodal VLA frameworks.

III. Methodology

A. NVIDIA GR00T Architecture

The original GR00T architecture contains two major components: the EAGLE Vision-Language Model, which tokenizes the language instruction and encodes the image observation; and the flow-matching-based Diffusion Transformer (DiT), which iteratively denoises input actions and proprioception tokens into a sequence of 16 robot actions. In our instance, the VLM takes in both the wrist RGB camera observations as well as the externally mounted RGB camera observations, and the action head makes use of the tokenized current robot joint positions, generating a 16-action trajectory for each of the 6 robot joints plus the gripper position (7-DOF output). This architecture is shown in Figure 1.

In this project, we utilize the GR00T N1.5 pretrained model, and fine-tune it using a manually gathered custom dataset. We treat the Kinova manipulator arm as a new embodiment and adjust the original GR00T configurations accordingly. During inference, each image-text-proprio input generates 16 predicted actions for the robot arm, which are sequentially executed on the Kinova arm.

B. Data Collection and Training

User demonstrations of the instructed tasks are recorded and transformed into the LeRobot dataset format, which is used to train our instance of GR00T. An Xbox controller connected to the Kinova arm is used to teleoperate the arm in the X/Y/Z dimensions, as well as the gripper opening/closing positions. We conduct 2 types of experiments to demonstrate the capabilities of our scene-aware VLA model:

1. Pick and place cups on a table
   1.1. Case with occluded goal locations
   1.2. Case with random goal locations
2. Pick and place cups at a height

The episodes listed below are used in fine-tuning the original GR00T architecture, as well as the Vista3D network.

Category	No. Episodes	Sample Task Description
Pick-and-place	36	`Pick up the orange cup and place it on the black cross.`
Pick-and-place (varied height)	9	`Pick up the orange cup and place it on top of the blue box.`

C. Custom Kinova Dataset with 3D Gaussian Representations

To support the integration of 3D perception, we will collect a new dataset using the Kinova Gen3 arm at the CMU AI Makerspace. Our setup will employ a single calibrated RGB-D camera mounted to capture both the workspace and the robot manipulations. The recorded RGB-D sequences will be used to reconstruct each scene as a 3D Gaussian Splat representation.

The dataset will feature pick-and-place tasks involving common kitchen items such as **cups, bowls, and plates**, captured under varied lighting, clutter, and occlusion conditions. The resulting data will be used both to train the 3D vision head and to

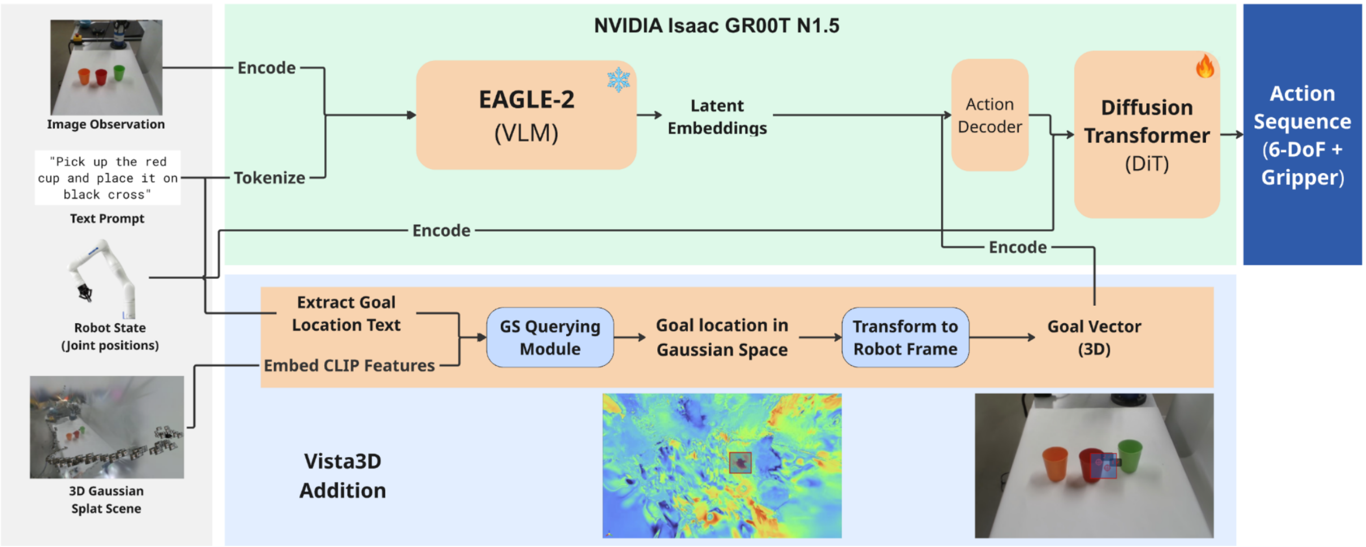

Fig. 2. Vista3D Pipeline. Vista3D augments GR00T by injecting an explicit 3D goal vector–derived from the Gaussian Splat scene–into the VLM latent stream.

evaluate its generalization to new objects and scene configurations.

D. 3D Gaussian Splatting Generation

Before training or evaluating the Visual Language Action model, it is essential to construct a static Gaussian Splat representation of the workspace that encodes the full geometric structure of the environment. The objective is to obtain a dense and spatially consistent set of Gaussian primitives that approximate the scene geometry and appearance from a continuous distribution. To achieve this, we first acquire a calibrated multiview dataset. The overhead RGB camera is rigidly mounted at a known pose relative to the robot base frame, and we compute the corresponding extrinsic transform

$$^{b}T_c \in SE(3)$$

through standard hand–eye calibration. Once this reference configuration is established, the camera is manually rotated around the workspace while maintaining its optical axis directed toward the set of target objects so that the collected images provide sufficient parallax for accurate structure recovery.

Following data capture, we process the image set using the standard COLMAP reconstruction pipeline. Feature extraction is performed using a scale invariant keypoint detector and descriptor to compute sets of features

$$\{f_i\}_{i=1}^{N}$$

for each image. Sequential matching is then applied to generate correspondences between temporally adjacent frames, yielding pairwise feature matches

$$\mathcal{M}_{ij} = \{(f_i^k, f_j^{k'})\}.$$

The global mapper uses these correspondences to solve the structure from motion problem by jointly optimizing camera poses

$$\{P_i\}$$

and sparse 3D points

$$\{X_j\}$$

via bundle adjustment. The resulting reconstruction provides accurate camera intrinsics, extrinsics, and a sparse point cloud that defines the geometric scaffold of the scene. The image to image transform generator is subsequently used to compute all relative view transformations that will later be leveraged during Gaussian optimization.

Once the calibrated scene geometry is available, we employ Splatfacto within the Nerfstudio framework to generate a high fidelity Gaussian Splat model. Splatfacto initializes a continuous field of anisotropic Gaussians

$$\mathcal{G} = \{g_k = (\mu_k, \Sigma_k, \alpha_k, c_k)\}$$

where each Gaussian is parameterized by its mean position, covariance, opacity, and color. These parameters are then jointly optimized using differentiable rendering to minimize the reprojection error across all collected images. The output is a complete set of configuration files, a dense Gaussian point cloud representation, and the final static map.

To incorporate semantic priors into this geometric map, we embed each Gaussian using a pretrained CLIP image encoder. For a Gaussian with associated appearance feature I_k, we compute its semantic vector

$$e_k = \text{CLIP}(I_k)$$

which is then attached to its corresponding spatial primitive. This produces a semantically enriched Gaussian field that can be queried during downstream tasks. The resulting representation is fed into the Gaussian querying module, enabling robust goal inference even when the target is absent from the robot's current RGB observation.

E. 3D Gaussian Querying Module

To provide GR00T with 3D spatial understanding, we introduce a lightweight querying module that identifies the goal object directly from the Gaussian Splat representation of the scene. The process is shown in the bottom part of Fig. 2 and consists of the following steps:

1) ***Language and Gaussian Matching:*** From the instruction, we extract the phrase that describes the goal object or target location (e.g., "place on black cross"). This phrase is encoded using the same text encoder as SplatMOVER [9], ensuring it lies in the same embedding space as the Gaussian features. Since each Gaussian primitive contains a CLIP-based feature vector, we compute its similarity to the encoded text and rank all Gaussians accordingly. The top ten most similar Gaussians are selected as candidates.

2) ***Camera-Frame Filtering:*** These candidate Gaussians are then transformed into the tripod camera coordinate frame to verify that the camera can actually observe them. Any candidates that fall behind the camera or outside the camera's field of view are removed. This step reduces false positives and ensures that only visible, physically plausible targets are considered.

3) ***Selecting the Final Target:*** From the remaining visible candidates, we select the one with the highest similarity to the text description. The 3D position associated with this Gaussian—an (x, y, z) coordinate in the reconstruction frame—is then transformed into the robot base frame so that the system can act on it directly.

4) ***Integration with GR00T:*** The resulting (x, y, z) target point is encoded using a small MLP and concatenated with GR00T's standard inputs (vision embeddings and proprioceptive tokens). This gives the action decoder an explicit 3D grounding of the goal object, rather than relying solely on 2D image features.

Unlike traditional 2D-only VLA models, GR00T lacks explicit 3D grounding and therefore cannot infer the location of a goal object when it is outside the camera's current viewpoint or partially occluded. The Gaussian Splat scene acts as a persistent, viewpoint-invariant memory of the environment, allowing the system to retrieve object positions even when they are not visible in the live RGB frame. By querying this semantically enriched 3D map, Vista3D provides GR00T with **explicit geometric goal information** that is not accessible from 2D images alone.

IV. Results and Evaluation

TABLE I
Comparison of success rates across different VLA configurations and visibility conditions.

VLA	Condition	# Trials	Successes	Success Rate
Vanilla GR00T	Visible	9	5	56%
Vanilla GR00T	Occluded	9	1	11%
Vanilla GR00T	Updated	9	3	33.33%
Vanilla GR00T	Height	9	0	0%
Vista3D (Ours)	**Occluded**	**9**	**4**	**44%**
Vista3D (Ours)	**Updated**	**12**	**5**	**41.6%**
Vista3D (Ours)	**Height**	**9**	**1**	**11%**

Performance Under Occlusion

The vanilla GR00T model achieves only **11%** success when the goal cup is not visible in the current camera frame. This is expected, as GR00T relies purely on instantaneous RGB observations: once the target is occluded, no visual embedding contains information about its location.

In contrast, Vista3D achieves **44%** success in the same setting. The improvement arises because the Gaussian Splat map encodes a *persistent, viewpoint-invariant* representation of the entire scene. The querying module retrieves the 3D location of the target from this global map even when the target is not visible in the live image. This confirms that adding an explicit 3D memory structure meaningfully improves manipulation robustness under partial observability.

Updated or Random Goal Locations

For tasks where the goal location changes between episodes (e.g., ”place the cup on the black cross” but with a different cross position each trial), Vista3D again outperforms GR00T (**41.6%** vs. **33.33%**). The Gaussian representation provides more stable geometric cues across runs, helping the robot disambiguate spatial relationships that vary episode-to-episode.

However, both models still fail in many trials. This is because the goal region is not a single canonical point in space but an area on the table. Since the current querying module returns the *single most similar Gaussian*, the resulting target location is often noisy or slightly misaligned with the true desired placement region.

Height-Based Placement Tasks

Both models perform poorly in tasks requiring height reasoning, though Vista3D improves from **0%** to **11%**. This small gain again comes from better 3D geometric grounding, but the overall failure mode is consistent:

The querying module reduces a potentially extended 3D surface (e.g., the *top* of a box) into a single coordinate. Height-based placements require predicting a *distribution* or *surface manifold*, not a point. Because the height target is not uniquely defined and depends on object geometry, a single-point output is insufficient for reliable execution. This explains why improvement exists but is modest.

Qualitative Analysis of Successes and Failures

Qualitatively, Vista3D succeeds in situations where:

- the target cup is temporarily occluded,
- the scene geometry is consistent across views,
- the target corresponds to a well-defined, localized 3D point.

Common failure cases include:

- selecting a Gaussian on the *edge* of the intended surface, causing misplacement,
- errors in CLIP-based semantic matching when objects have similar appearances,
- height tasks where the ”goal location” is a 3D *region* instead of a point.

We have included some videos for our demonstrations in the link provided in the footnote.[1]

V. Conclusion & Future Work

Our results demonstrate that introducing 3D scene representations into VLA architectures provides clear benefits in settings where 2D vision alone is insufficient. Vista3D consistently outperforms the vanilla GR00T model in scenarios involving occlusion or ambiguous visual cues, confirming that an explicit 3D memory of the environment enables more reliable goal localization. The approach is particularly effective when the goal corresponds to a well-defined point in space, where the querying module can retrieve a stable and geometrically meaningful target.

However, the current design also exposes important limitations. Because the querying module returns a *single* 3D coordinate, the model struggles with tasks where the desired placement corresponds to an extended 3D region or surface (e.g., the top of a box). This point-based formulation prevents the model from leveraging the full richness of the Gaussian Splat representation and underutilizes the spatial and semantic structure encoded in the map. To fully capitalize on 3D information, future work will move beyond point querying toward an architecture that *encodes the entire Gaussian scene as a sequence of tokens*. Instead of treating Gaussian features as an auxiliary lookup table, we aim to integrate them directly into the VLM, allowing the

[1] **Experiment Videos**: https://drive.google.com/drive/folders/1mGzrLh-Dgj0xu-AQqN5SEFjhsjDtkq85?usp=drive_link

network to jointly reason over geometry, language, and visual context in an end-to-end manner. The proposed extension (shown in Fig. 3) introduces full-scene tokens and task-specific sparsifier tokens, enabling richer 3D grounding and more expressive action generation. This direction aligns with recent trends in scene-centric multimodal models and offers a promising path toward more robust and generalizable robotic manipulation.

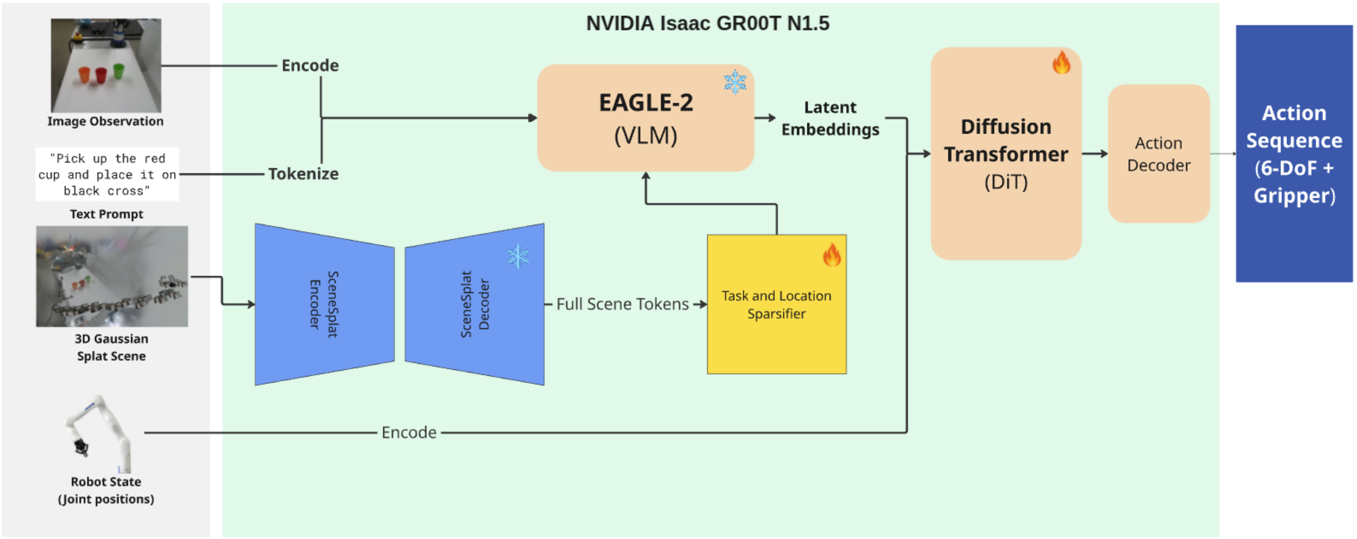

Fig. 3. Modified Vista3D Pipeline. In this extended version, full scene tokens and task-specific sparsifier tokens are passed to the VLM, enabling end-to-end 3D scene encoding rather than single-point querying.

REFERENCES

[1] J. Bjorck, et al., "GR00T N1: An Open Foundation Model for Generalist Humanoid Robots," *arXiv preprint arXiv:2503.14734*, 2025.

[2] D. Driess, et al., "PaLM-E: An Embodied Multimodal Language Model," in *Proc. 40th Int. Conf. on Machine Learning (ICML)*, vol. 202, pp. 8469–8488, 2023.

[3] B. Zitkovich, et al., "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *Proc. Conf. on Robot Learning (CoRL)*, PMLR, 2023.

[4] Shukor, Mustafa, et al. "Smolvla: A vision-language-action model for affordable and efficient robotics." arXiv preprint arXiv:2506.01844 (2025).

[5] Jiaming Liu, Hao Chen, et al. "HybridVLA-Collaborative Diffusion and Autoregression in a Unified Vision-Language-Action Model" *arXiv preprint arxiv:2503:10631*, 2025

[6] Y. Yuan, et al., "3D Diffusion Policy: Learning Visuomotor Skills from 3D Point Clouds," *arXiv preprint arXiv:2310.16828*, 2023.

[7] Y. Yuan, et al., "Improved 3D Diffusion Policy: Scalable Visuomotor Learning with Efficient 3D Representations," *arXiv preprint arXiv:2403.09812*, 2024.

[8] F. Liu, et al., "LEGaussians: Learning Gaussian Representations for Efficient 3D Reconstruction in Robotics," *arXiv preprint arXiv:2405.11072*, 2024.

[9] S. Wang, et al., "SplatMover: Real-Time 3D Gaussian Splatting for Robotic Manipulation and Motion Capture," *arXiv preprint arXiv:2408.01456*, 2024.

[10] Steiner, Remo, et al. "mindmap: Spatial Memory in Deep Feature Maps for 3D Action Policies." arXiv preprint arXiv:2509.20297 (2025).

[11] Kerbl, Bernhard, et al. "3D Gaussian splatting for real-time radiance field rendering." ACM Trans. Graph. 42.4 (2023): 139-1.

[12] Y. Wang, M. Zhang, Z. Li, T. Kelestemur, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li, "D³Fields: Dynamic 3D Descriptor Fields for Zero-Shot Generalizable Rearrangement," in *Proc. Conf. on Robot Learning (CoRL)*, 2024.

[13] Lee, Jason, et al. "Molmoact: Action reasoning models that can reason in space." arXiv preprint arXiv:2508.07917 (2025).

[14] Li, Yue, et al. "Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining." arXiv preprint arXiv:2503.18052 (2025).

[15] Halacheva, Anna-Maria, et al. "GaussianVLM: Scene-centric 3D Vision-Language Models using Language-aligned Gaussian Splats for Embodied Reasoning and Beyond." arXiv preprint arXiv:2507.00886 (2025).