# GAUSSIANDREAM: A FEED-FORWARD 3D GAUSSIAN WORLD MODEL FOR ROBOTIC MANIPULATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision-language-action (VLA) policies have shown strong performance in language-conditioned robotic manipulation, but standard action imitation provides limited supervision for learning explicit 3D geometry and future scene evolution. We propose **GaussianDream**, a feed-forward 3D Gaussian world model that introduces structured spatial and temporal supervision into VLA policy learning. GaussianDream inserts a set of learnable queries into the policy encoder to form a latent GaussianDream prefix. During training, the prefix is decoded by a current reconstruction head and a future prediction head to model the observed 3D scene and its short-horizon evolution. The current branch is supervised with RGB rendering and pseudo depth, while the future branch additionally uses future RGB, pseudo depth, and pseudo 3D scene-flow supervision. At inference time, all auxiliary reconstruction and prediction heads are removed, and only the learned GaussianDream prefix is retained for action generation, requiring no test-time Gaussian reconstruction or future rollout. GaussianDream achieves **98.4%** success on LIBERO, **87.0%** on LIBERO-Plus, and **54.8%** on RoboCasa Human-50. We further validate GaussianDream on real robots using both single-arm and dual-arm platforms, achieving **50.0%** average success on the single-arm tasks and **40.0%** success on the higher-precision dual-arm manipulation task.

## 1 INTRODUCTION

General-purpose robotic manipulation requires an agent to interact with its environment and accomplish language-specified goals, making it a fundamental problem in embodied intelligence. Recent Vision-Language-Action (VLA) models, such as RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024), $\pi_0$ (Black et al., 2024), and $\pi_{0.5}$ (Physical Intelligence et al., 2025), have achieved substantial progress by transferring semantic knowledge from pretrained Vision-Language Models (VLMs) to action generation. Despite this progress, reliable spatial perception remains a challenge for VLA policies. In **2D VLA policies**, geometric information such as depth, relative spatial relationships, and 3D object structure is often encoded only implicitly in pretrained visual representations (Sun et al., 2025; Qian et al., 2025; Song et al., 2026; Li et al., 2025b;a). Imprecise spatial understanding can lead to manipulation errors, such as inaccurate grasp locations or object placements.

Beyond understanding the current scene, manipulation involves changes in object poses and spatial relationships as the robot interacts with its environment. Learning to predict these short-horizon changes can encourage policy representations to capture how scene geometry evolves during an interaction. Meanwhile, consecutive visual observations in robot trajectories contain rich spatial and temporal information, from which dense appearance, depth, and motion supervision can be derived. However, standard VLA training mainly optimizes action prediction (Black et al., 2024; Physical Intelligence et al., 2025); these task-relevant targets do not explicitly require reconstructing scene structure or predicting its evolution. Dense supervision from robot observations can supply these additional spatial and temporal constraints.

Several complementary lines of research have sought to address these challenges. **3D-enhanced VLAs** inject depth, point clouds, or other 3D representations to improve spatial perception (Sun et al., 2025; Deng et al., 2025; Zhou et al., 2025; Ni et al., 2025). In parallel, **video/latent world models** learn scene dynamics by predicting future observations or latent states (Ye et al., 2026; Kim et al., 2026; Li et al., 2026a; Bi et al., 2025; Cen et al., 2025; Yuan et al., 2026). Dense supervision in VLA



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

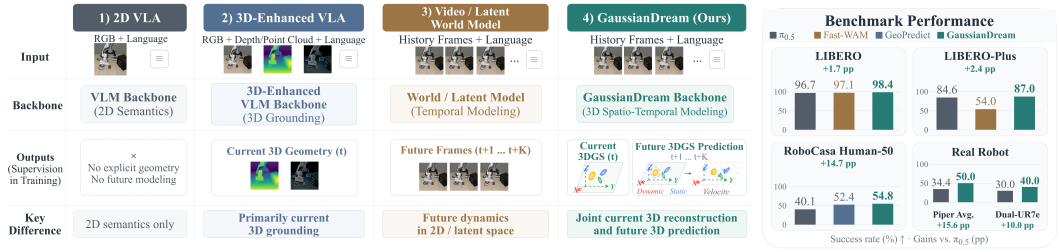


Figure 1: **Comparison of manipulation policy paradigms and benchmark performance. Left:** GaussianDream uses current Gaussian reconstruction and future prediction as structured 3D supervision during training, while retaining prefix-based action generation without test-time Gaussian decoding or rollout. **Right:** Success rates (%) compared with representative baselines on LIBERO, LIBERO-Plus, RoboCasa Human-50, Piper tasks, and a dual-UR7e precision task. Improvements over $\pi_{0.5}$ are reported in percentage points (pp).

training, including depth and geometric objectives, provides additional spatial constraints beyond action prediction (Li et al., 2025b;a; Song et al., 2026). However, recovering current geometry alone does not specify how it changes, while future prediction in visual or latent spaces need not establish explicit 3D correspondence to the current scene. Together, these advances raise a central question: how can dense supervision of scene structure and its evolution jointly shape the representations used for action generation? Addressing this question calls for a shared 3D representation that explicitly relates the current scene to its future changes and connects both to observable training targets.

To this end, we present **GaussianDream**, a **unified 3D Gaussian world model** that integrates **3D spatial information injection, future prediction, and dense supervision** within a single network for language-conditioned robotic manipulation. The overall framework is shown in Fig. 2, while Fig. 1 compares GaussianDream with other manipulation policy paradigms. 3D Gaussians jointly encode geometry and appearance in a renderable scene representation (Kerbl et al., 2023). By reconstructing the current scene and predicting short-horizon displacements of the same Gaussian primitives, GaussianDream expresses future changes relative to a shared current state. Differentiable rendering and geometric supervision constrain both states, coupling spatial and temporal learning in the representation used for action generation.

Specifically, we introduce a set of learnable GaussianDream Queries into the policy encoder to extract 3D-aware representations from temporal observations. These queries form a GaussianDream prefix shared by action generation and the auxiliary world-model decoders. A static Gaussian head reconstructs the current scene, while a dynamic Gaussian head predicts displacements of its Gaussian centers in subsequent frames. RGB rendering losses and pseudo depth supervise both current and future states, while pseudo 3D scene-flow targets additionally constrain the predicted displacements. During policy learning, these objectives are optimized jointly with the action loss, allowing spatial structure and short-horizon motion to shape the features used for control. At inference time, the auxiliary Gaussian heads are removed, while the temporal encoder and learned GaussianDream prefix are retained for action generation without scene decoding or future rollout.

GaussianDream achieves state-of-the-art performance among the compared methods on RoboCasa Human-50 (Nasiriany et al., 2024) with **54.8%** success, improving over $\pi_{0.5}$ by 14.7 percentage points, and competitive results on LIBERO (Liu et al., 2023) with **98.4%** success. Under zero-shot transfer to LIBERO-Plus (Fei et al., 2025), it achieves the highest overall success rate among the evaluated methods at **87.0%**, exceeding $\pi_{0.5}$ by 2.4 percentage points. Real-robot experiments further demonstrate its effectiveness, with **50.0%** average success on single-arm tasks and **40.0%** success on the dual-arm task. Ablation studies support the effectiveness of current reconstruction, future prediction, and dense supervision for policy learning.

In summary, our main contributions are:

- **Unified 3D Gaussian World Model.** GaussianDream integrates 3D spatial information injection, future prediction, and dense supervision within a single network for VLA policy learning.
- **Shared Gaussian Representation for Geometry and Motion.** Current reconstruction and displacement prediction over the same Gaussian primitives establish explicit correspondence for learning geometry and motion in a shared policy representation.

- **Efficient Training-Inference Design.** Auxiliary Gaussian heads supervise training, while deployment conditions actions on the learned GaussianDream prefix without scene decoding.
- **Extensive Empirical Validation.** Experiments demonstrate strong performance on standard benchmarks, zero-shot transfer, and real robots. Ablations and further analyses examine component effectiveness, learned spatial representations, the relationship between world-model quality and policy success, and inference efficiency.

## 2 Related Work

**Vision-language robot policies.** Vision-language-action (VLA) models adapt pretrained vision-language representations to robot control, enabling language-conditioned manipulation across diverse tasks. RT-1 (Brohan et al., 2022) and RT-2 (Brohan et al., 2023) demonstrate the benefit of scaling robot data and transferring vision-language priors to action generation. Octo (Octo Model Team et al., 2024), OpenVLA (Kim et al., 2024), $\pi_0$ (Black et al., 2024), and $\pi_{0.5}$ (Physical Intelligence et al., 2025) further advance open-vocabulary instruction following, continuous action modeling, and generalist policy learning. Recent works improve action tokenization, deployability, speech conditioning, force awareness, and mobile manipulation (Pertsch et al., 2025; Wen et al., 2024; Shukor et al., 2025; Zhao et al., 2025; Yu et al., 2026; Wu et al., 2025). These developments have substantially broadened the scope and practicality of VLA policies across robot platforms and interaction settings. However, most improvements primarily target action generation, model efficiency, or broader task conditioning rather than explicitly constraining the learned representation to encode scene geometry and its temporal evolution. While these models provide strong semantic priors, their training mainly focuses on action prediction, leaving geometric information in robot observations only indirectly encoded in the learned visual representations. GaussianDream preserves the standard VLA action interface while introducing explicit 3D reconstruction and future geometric prediction as additional training supervision for learning spatially grounded representations.

**3D-enhanced manipulation policies.** 3D Diffusion Policy (DP3) (Ze et al., 2024) uses compact point-cloud representations for generalizable visuomotor policy learning, while FP3 develops a 3D foundation policy based on point-cloud observations. GeoVLA (Sun et al., 2025), StereoVLA (Deng et al., 2025), VLA-4D (Zhou et al., 2025), SwiftVLA (Ni et al., 2025), BridgeVLA (Li et al., 2026b), and Any3D-VLA (Fan et al., 2026) incorporate depth, point clouds, 4D features, or projected 3D representations to improve spatial perception. QDepth-VLA (Li et al., 2025b), Spatial Forcing (Li et al., 2025a), GeoAware-VLA (Abouzeid et al., 2025), and VGA (Song et al., 2026) further explore depth supervision, geometric regularization, and geometry-aware representations. More recently, LaMP (Wang et al., 2026) introduces 3D scene flow as a latent motion prior for VLA policy learning, while RoboFlow4D (Lin et al., 2026) predicts multi-frame 3D scene flow to model future geometric motion for robotic manipulation. GeoPredict (Qian et al., 2025) further introduces predictive geometry through future keypoint trajectories and depth-supervised Gaussian prediction. These methods demonstrate the benefit of explicit geometry for spatial reasoning and manipulation. However, existing approaches either focus primarily on 3D representations of the current observation, which may discard part of the original scene information during geometric abstraction, or model future geometric prediction as a separate motion prior or auxiliary branch in a representation space different from that of the current scene. As a result, current 3D state modeling and future geometric prediction are not represented in a unified space. In contrast, GaussianDream unifies current 3D reconstruction and short-horizon future geometric prediction within the same 3D Gaussian space, while leveraging richer supervision from RGB, pseudo depth, and pseudo 3D scene flow during training.

**World models for robotic manipulation.** DreamZero (Ye et al., 2026), Cosmos Policy (Kim et al., 2026), LingBot-VA (Li et al., 2026a), Motus (Bi et al., 2025), WorldVLA (Cen et al., 2025), and Fast-WAM (Yuan et al., 2026) model future visual observations or latent representations together with action generation to improve policy learning and generalization. Most of these approaches perform future prediction in 2D visual or latent spaces, while generative video rollout can introduce additional inference cost. GWM (Lu et al., 2025) further develops a generative 3D Gaussian world model for future-scene prediction and can serve as a neural simulator for policy learning. In contrast, GaussianDream performs current-scene reconstruction and short-horizon future prediction within a unified 3D Gaussian space, while the learned GaussianDream prefix directly conditions action

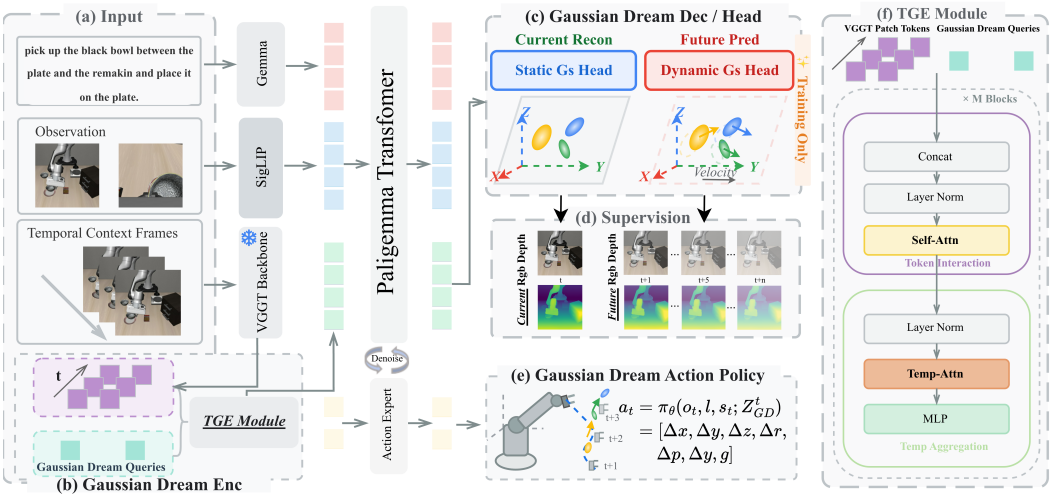


Figure 2: Overview of **GaussianDream**. During training, temporal visual features and learnable GaussianDream queries produce a compact prefix, which is decoded into current and future 3D Gaussian states for RGB, pseudo depth, and pseudo 3D scene-flow supervision. During inference, all auxiliary Gaussian decoding heads are discarded, and only the prefix conditions action generation.

generation. At inference time, the auxiliary Gaussian reconstruction and prediction heads are removed, leaving only the GaussianDream prefix for policy inference.

# 3 Method

## 3.1 Overview of GaussianDream

Fig. 2 illustrates the overall architecture. Let $\mathbf{o}_t$ denote the current multi-view observation, $\mathbf{o}_{t-K:t}$ a short temporal history of the agent-view observations, $\mathbf{l}$ the language instruction, and $\mathbf{s}_t$ the robot state. GaussianDream preserves the action interface of the base continuous-action policy, but augments its multimodal context with learned prefix tokens $\mathbf{Z}_t^{\mathrm{GD}}$. The GaussianDream prefix is constructed from the temporal agent-view observation sequence, which provides consistent spatial context for 3D reconstruction and future prediction. The prefix is generated by a temporal 3D-aware encoder:

$$\mathbf{Z}_t^{\mathrm{GD}} = \mathcal{F}_\omega\left(\mathbf{o}_{t-K:t}, \mathbf{Q}_{\mathrm{GD}}\right), \tag{1}$$

where $\mathbf{Q}_{\mathrm{GD}}$ are learnable GaussianDream queries and $\mathcal{F}_\omega$ is implemented by the Temporal Gaussian Evolution (TGE) module in Sec. 3.3. Image, language, and GaussianDream tokens share the 2048-dimensional PaliGemma/Gemma-2B prefix space, while the action expert uses a 1024-dimensional hidden space. We use three context frames, $\{t-10, t-5, t\}$, to provide sparse motion cues.

During training, the same prefix is decoded by two auxiliary Gaussian heads:

$$\mathcal{G}_t = \mathcal{R}_\phi(\mathbf{Z}_t^{\mathrm{GD}}, \mathbf{o}_t), \qquad \hat{\mathcal{G}}_{t+\Delta} = \mathcal{D}_\psi(\mathcal{G}_t, \mathbf{Z}_t^{\mathrm{GD}}, \Delta). \tag{2}$$

Here $\mathcal{R}_\phi$ maps the prefix to a renderable current Gaussian state $\mathcal{G}_t$, and $\mathcal{D}_\psi$ predicts horizon-conditioned future Gaussian changes. These decoders serve only as training-time supervisory interfaces. The policy conditions on the original multimodal inputs and the GaussianDream prefix:

$$\mathbf{a}_t = \pi_\theta\left(\mathbf{o}_t, \mathbf{l}, \mathbf{s}_t; \mathbf{Z}_t^{\mathrm{GD}}\right). \tag{3}$$

Thus, reconstruction and future prediction supervise GaussianDream prefix during training. During inference, no Gaussian decoding is performed, and the prefix directly conditions action generation.

## 3.2 Current Gaussian Reconstruction

The current reconstruction branch provides explicit 3D spatial grounding by requiring the Gaussian-Dream prefix to represent a renderable 3D scene state. The learned prefix $\mathbf{Z}_t^{\mathrm{GD}}$ serves as a compact

3D Gaussian latent representation that captures both current scene geometry and short-horizon interaction dynamics. Its temporal evolution capability will be further modeled in Sec. 3.3.

We reshape the 1024 GaussianDream tokens into a spatially organized $32 \times 32$ latent grid, allowing the prefix to preserve spatial locality and dense geometric structure. A Gaussian decoder then upsamples this latent grid into a dense feature map:

$$\mathbf{F}_t^{\mathrm{G}} = \mathcal{B}_{\mathrm{G}}\left(\mathrm{Grid}(\mathbf{Z}_t^{\mathrm{GD}})\right), \qquad \mathbf{F}_t^{\mathrm{G}} \in \mathbb{R}^{256\times256\times128}. \tag{4}$$

Here, the $256 \times 256$ spatial resolution corresponds to dense Gaussian layout prediction, while the 128-dimensional feature channels encode shared geometric and appearance information for subsequent Gaussian attribute estimation.

Lightweight prediction heads are then applied to construct the current Gaussian scene state. The geometry head predicts depth, rotation, scale, and opacity, while the appearance head conditions on the current agent-view RGB observation $\mathbf{o}_t$ to estimate degree-1 spherical harmonics coefficients for Gaussian appearance:

$$\mathbf{D}_t, \Theta_t^{\mathrm{geo}} = \mathcal{H}_{\mathrm{geo}}(\mathbf{F}_t^{\mathrm{G}}), \qquad \Theta_t^{\mathrm{app}} = \mathcal{H}_{\mathrm{app}}(\mathbf{F}_t^{\mathrm{G}}, \mathbf{o}_t). \tag{5}$$

We unproject $\mathbf{D}_t$ into Gaussian centers $\boldsymbol{\mu}_i^t$ and predict the corresponding Gaussian attributes $\theta_i^t$:

$$\mathcal{G}_t = \mathcal{A}(\mathbf{D}_t, \Theta_t) = \{(\boldsymbol{\mu}_i^t, \theta_i^t)\}_{i=1}^{N_t}, \tag{6}$$

where $\Theta_t = (\Theta_t^{\mathrm{geo}}, \Theta_t^{\mathrm{app}})$ and $N_t = 256 \times 256$.

The reconstructed Gaussian scene is rendered for RGB and pseudo depth supervision. By constraining the prefix to reconstruct explicit 3D geometry, this branch injects dense spatial structure into the policy representation and provides the static Gaussian template for future evolution prediction.

### 3.3 Future Gaussian Prediction

Current reconstruction specifies the scene state at time $t$, but does not supervise how the scene evolves after interaction. To expose dense temporal structure in robot videos, GaussianDream predicts short-horizon Gaussian evolution in structured 3D space.

Given $\mathbf{o}_{t-K:t}$, we extract multi-scale 3D-aware patch features using VGGT (Wang et al., 2025). Since VGGT learns strong static 3D priors through global frame attention but provides relatively weak supervision for temporal interaction dynamics, we introduce learnable GaussianDream queries to aggregate spatial geometry while modeling future evolution across time. For each context frame, VGGT features are pooled into a $32 \times 32$ grid and projected into temporal tokens:

$$\mathbf{P}_{t-K:t}^{(m)} = W_m\, \mathcal{P}_{32\times32}\left(\mathcal{E}_{\mathrm{VGGT}}^{(m)}(\mathbf{o}_{t-K:t})\right), \tag{7}$$

where $m$ indexes the feature scale.

The Temporal Gaussian Evolution (TGE) module lets temporal tokens interact with learnable GaussianDream queries. The 2048-dimensional queries are first projected into a 512-dimensional temporal space, fused with VGGT features from multiple frames, and finally projected back to the prefix space. TGE contains 12 attention blocks with 8 heads, alternating frame-wise spatial interaction and time-slot temporal attention to jointly capture current 3D structure and future geometric evolution. The current-step output is used as the GaussianDream prefix:

$$\mathbf{Z}_t^{\mathrm{GD}} = \mathrm{Proj}_{512\to2048}\left[\mathrm{TGE}\left(\mathrm{Proj}_{2048\to512}(\mathbf{Q}_{\mathrm{GD}}), \{\mathbf{P}_{t-K:t}^{(m)}\}_{m=1}^{M}\right)_t\right]. \tag{8}$$

Given $\mathcal{G}_t = \{(\boldsymbol{\mu}_i^t, \theta_i^t)\}_{i=1}^{N_t}$, the dynamic head predicts horizon-conditioned center displacements:

$$\begin{aligned} \boldsymbol{\nu}_t^{(\Delta)} &= \mathcal{H}_{\mathrm{vel}}\left(\mathcal{B}_{\mathrm{pred}}(\mathbf{Z}_t^{\mathrm{GD}}), \mathbf{e}_\Delta\right), \qquad \Delta\mathbf{X}_t^{(\Delta)} = \alpha_\Delta \boldsymbol{\nu}_t^{(\Delta)}, \\ \hat{\boldsymbol{\mu}}_i^{\,t+\Delta} &= \boldsymbol{\mu}_i^t + \Delta\mathbf{x}_i^{(\Delta)}, \qquad \hat{\mathcal{G}}_{t+\Delta} = \left\{\left(\hat{\boldsymbol{\mu}}_i^{\,t+\Delta}, \theta_i^t\right)\right\}_{i=1}^{N_t}. \end{aligned} \tag{9}$$

Here $\mathbf{e}_\Delta$ is a learnable horizon embedding and $\alpha_\Delta$ is a temporal scale factor. Since robot interactions exhibit non-uniform motion patterns and time-dependent uncertainty, the horizon embedding $\mathbf{e}_\Delta$ enables the model to distinguish different dynamic modes across prediction horizons. The future state reuses non-positional attributes $\theta_i^t$, focusing prediction on interaction-induced geometric changes while preserving the current Gaussian template. We supervise horizons from $t+1$ to $t+5$.

### 3.4 GaussianDream Training and Efficient Inference

GaussianDream follows a two-stage and asymmetric training–inference strategy. In Stage I, the model learns spatial perception and short-horizon future prediction through 3D Gaussian reconstruction and prediction, allowing the GaussianDream prefix to encode current 3D structure and future geometric changes. In Stage II, the learned GaussianDream prefix is jointly optimized with the action policy so that it can better support action generation. At inference time, the auxiliary Gaussian decoding branches are removed, and only the GaussianDream prefix is retained for policy conditioning.

**Stage I: GaussianDream pretraining.** We first pretrain the GaussianDream representation without action learning. For each robot demonstration sequence, RGB observations are paired with pseudo depth and pseudo 3D scene-flow targets constructed from adjacent frames. This stage jointly optimizes current-scene reconstruction and short-horizon future prediction:

$$\mathcal{L}_{\mathrm{GD}} = \underbrace{\lambda_{\mathrm{cur}}^{\mathrm{depth}}\mathcal{L}_{\mathrm{cur}}^{\mathrm{depth}} + \lambda_{\mathrm{cur}}^{\mathrm{render}}\mathcal{L}_{\mathrm{cur}}^{\mathrm{render}}}_{\mathcal{L}_{\mathrm{cur}}} + \underbrace{\sum_{\Delta\in\mathcal{H}} w_\Delta \left(\lambda_{\mathrm{depth}}\mathcal{L}_{\mathrm{depth}}^{(\Delta)} + \lambda_{\mathrm{render}}\mathcal{L}_{\mathrm{render}}^{(\Delta)} + \lambda_{\mathrm{flow}}\mathcal{L}_{\mathrm{flow}}^{(\Delta)}\right)}_{\mathcal{L}_{\mathrm{fut}}}. \tag{10}$$

The current branch is supervised by RGB rendering and pseudo depth, while the future branch additionally uses future RGB, pseudo depth, and pseudo 3D scene-flow supervision. The flow loss constrains the predicted Gaussian center displacements to match valid pseudo 3D scene-flow targets. Through these objectives, the GaussianDream prefix learns both the current 3D spatial structure and short-horizon future geometric changes.

**Stage II: GaussianDream action policy learning.** After pretraining, we jointly optimize the GaussianDream representation and the action policy. Let $\mathbf{c}_t = (\mathbf{o}_t, \mathbf{l}, \mathbf{s}_t; \mathbf{Z}_t^{\mathrm{GD}})$ denote the policy condition. The action objective is:

$$\mathcal{L}_{\mathrm{act}} = \mathbb{E}_{\tau,\boldsymbol{\epsilon},\mathbf{a}_t^*}\left[\left\|\mathbf{v}_\theta\left(\tau\boldsymbol{\epsilon} + (1-\tau)\mathbf{a}_t^*, \mathbf{c}_t, \tau\right) - (\boldsymbol{\epsilon} - \mathbf{a}_t^*)\right\|_2^2\right], \tag{11}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $\tau$ follows the flow-matching time distribution. The overall objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{GD}}\mathcal{L}_{\mathrm{GD}}. \tag{12}$$

The action objective adapts the GaussianDream prefix to robot control, while the auxiliary Gaussian objectives preserve the 3D spatial and future-evolution information learned in Stage I.

**Inference.** At inference time, we cache three temporal observations $\{t-10, t-5, t\}$ and encode them into the GaussianDream prefix. The prefix is then directly injected into the policy context to condition action generation. All auxiliary current-reconstruction and future-prediction branches are removed, and no Gaussian decoding is performed.

## 4 Experiments

We first introduce the experimental setup (Sec. 4.1). Next, we compare GaussianDream with representative VLA, 3D-enhanced, and world-model baselines across simulation and real-world manipulation benchmarks (Sec. 4.2). We then conduct ablation studies and further analyses to examine the contributions of individual components and the relationship between reconstruction quality and policy performance (Sec. 4.3). Finally, we qualitatively analyze GaussianDream's current reconstruction and future prediction capabilities, together with depth-aware spatial representations learned by different methods (Sec. 4.4).

### 4.1 Experimental Setup

**Simulation benchmarks.** We evaluate GaussianDream on LIBERO (Liu et al., 2023) and RoboCasa (Nasiriany et al., 2024). For LIBERO, we follow the Spatial, Object, Goal, and Long protocols with 50 demonstrations and 50 evaluation trials. For RoboCasa, we use the Human-50 few-shot setting over 24 long-horizon kitchen tasks, with 50 trials per task across five scenes. Together, these benchmarks cover spatial reasoning, object grounding, goal-conditioned manipulation, and long-horizon household tasks.

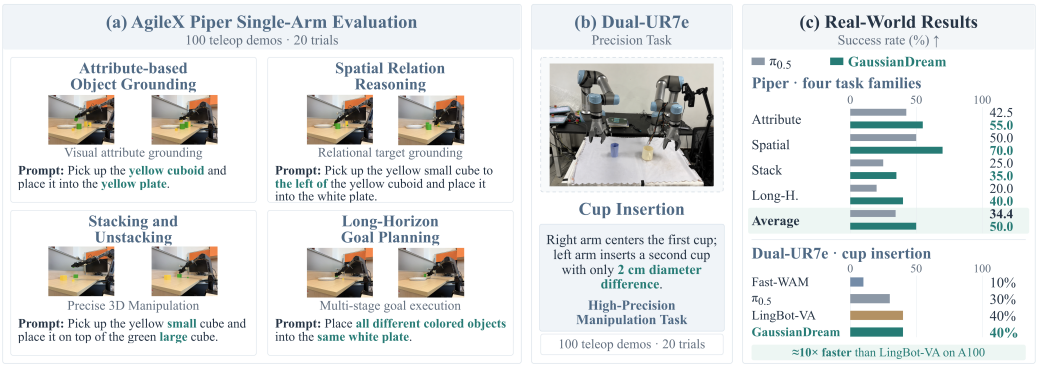


Figure 3: Real-world evaluation on the AgileX Piper and dual-UR7e platforms. We show representative tasks from four categories on the Piper platform, together with example language prompts and success rates. We also include a higher-precision dual-arm manipulation task on the dual-UR7e setup.

Table 1: Success rates (%) on LIBERO (Liu et al., 2023). GaussianDream achieves the best Spatial, Object, Goal, and average performance among representative VLA and 3D-enhanced methods.

| Method | Spatial | Object | Goal | Long | Avg. |
|---|---|---|---|---|---|
| $\pi_0$ (Black et al., 2024) | 96.8 | 98.8 | 95.8 | 85.2 | 94.1 |
| $\pi_{0.5}$ (Physical Intelligence et al., 2025) | 97.8 | 98.8 | 97.6 | 92.4 | 96.7 |
| GeoPredict (Qian et al., 2025) | 98.0 | 98.2 | 95.7 | 94.0 | 96.5 |
| QDepth-VLA (Li et al., 2025b) | 97.6 | 96.6 | 95.2 | 90.0 | 94.9 |
| GeoVLA (Sun et al., 2025) | 98.4 | 99.0 | 96.6 | 96.6 | 97.7 |
| Spatial Forcing (Li et al., 2025a) | 98.6 | 98.4 | 98.2 | 95.4 | 97.6 |
| VLA-4D (Zhou et al., 2025) | 97.9 | 98.6 | 97.8 | 94.8 | 97.4 |
| **GaussianDream** | **99.0** | **99.6** | **99.0** | 96.0 | **98.4** |

Table 2: Success rates (%) on RoboCasa Human-50 (Nasiriany et al., 2024). GaussianDream achieves the best average result and strongest Pick&Place performance among compared methods in this benchmark.

| Method | Pick&Place | Doors/Drawers | Others | Avg. |
|---|---|---|---|---|
| $\pi_0$ (Black et al., 2024) | 14.0 | 53.1 | 58.5 | 42.4 |
| $\pi_{0.5}$ (Physical Intelligence et al., 2025) | 36.0 | 46.5 | 39.5 | 40.1 |
| BC-Transformer (Nasiriany et al., 2024) | 3.8 | 46.7 | 38.0 | 28.8 |
| GWM (Lu et al., 2025) | 14.8 | 54.3 | 49.8 | 39.3 |
| GeoPredict (Qian et al., 2025) | 22.7 | **75.1** | **62.4** | 52.4 |
| Being-H0.5 (Luo et al., 2026) | 36.0 | 71.7 | 57.6 | 53.9 |
| **GaussianDream** | **43.8** | 66.3 | 54.4 | **54.8** |

**Real-robot evaluation.** We evaluate GaussianDream in both single-arm and dual-arm settings. On the AgileX Piper platform, we consider 12 tasks spanning four categories: Attribute-based Object Grounding, Spatial Relation Reasoning, Stacking and Unstacking, and Long-Horizon Goal Planning, with 4/4/2/2 tasks in each category, respectively. For each task, we collect 100 demonstrations and conduct 20 evaluation trials. We further evaluate GaussianDream on a dual-UR7e platform using a higher-precision task, providing additional validation of GaussianDream in a dual-arm setting.

### 4.2 Main Results

**LIBERO and LIBERO-Plus.** As shown in Tab. 1, GaussianDream achieves a 98.4% average success rate on LIBERO (Liu et al., 2023), outperforming representative VLA and 3D-enhanced baselines. It obtains the best results on Spatial (99.0%), Object (99.6%), and Goal (99.0%). We attribute these gains to the Gaussian world representation, which explicitly captures both scene appearance and geometry, while future prediction further models how the 3D scene evolves.

Since several recent VLA and 3D-enhanced methods do not provide publicly reproducible LIBERO-Plus evaluations, we focus the LIBERO-Plus comparison on methods with available results, including representative world-action models and LaMP, which incorporates 3D scene flow. As shown in Tab. 3, GaussianDream achieves the best overall average success rate of 87.0%, with strong robustness under camera, background, noise, and layout shifts. GaussianDream also requires only 531 ms per inference step on an A100 GPU, providing a favorable balance between robustness and inference efficiency. In the separate LIBERO-Long checkpoint comparison, GaussianDream achieves 79.7% success on LIBERO-Plus versus 74.3% for LingBot-VA, a 5.4-percentage-point gain.

**RoboCasa.** As shown in Tab. 2, GaussianDream achieves the best average success rate of 54.8% on RoboCasa (Nasiriany et al., 2024), with the strongest performance on Pick&Place tasks. Compared with the base $\pi_{0.5}$ policy, GaussianDream improves the average success rate by 14.7 percentage points and Pick&Place performance by 7.8 percentage points, demonstrating consistent gains over the base

Table 3: Comparison with representative world-action and motion-enhanced methods on LIBERO and LIBERO-Plus in terms of success rate (%), model size, and inference latency. **Bold** denotes the best result among standard non-privileged methods.

| | LIBERO | LIBERO-Plus | | | | | | | | Efficiency | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Avg. | Camera | Robot | Lang. | Light | BG. | Noise | Layout | Avg. | Params | A100 |
| **VLA Baseline** | | | | | | | | | | | |
| $\pi_{0.5}$ | 96.7 | 72.0 | 76.6 | 86.5 | 96.1 | 95.2 | 86.7 | 86.0 | 84.6 | 3.3B | **268 ms** |
| **World-Action and Motion-Enhanced Methods** | | | | | | | | | | | |
| Cosmos-Policy | **98.5** | 75.8 | 63.3 | 81.7 | 96.5 | 88.9 | 92.7 | 82.2 | 82.2 | 2B | 710 ms |
| DreamZero-Self-Adapt | 88.7 | 26.3 | 23.5 | 56.4 | 63.5 | 42.0 | 43.5 | 54.8 | 42.1 | 14B | 5616 ms |
| Fast-WAM | 97.1 | 25.6 | 42.9 | 73.9 | 61.7 | 54.9 | 43.6 | 66.8 | 54.0 | 6B | 700 ms |
| LaMP | 98.3 | 64.5 | 69.6 | 88.2 | 95.3 | 97.4 | 76.9 | 73.8 | 79.3 | 4B+ | – |
| **LIBERO-Long Checkpoint Comparison** | | | | | | | | | | | |
| LingBot-VA-Libero-Long | **98.5** | 44.2 | **84.5** | **93.0** | 80.7 | 73.4 | 65.5 | 87.2 | 74.3 | 5.3B | 5560 / 1220* ms |
| GaussianDream-Libero-Long | 96.0 | 54.6 | 71.3 | 87.5 | 95.4 | 93.0 | 81.2 | 85.8 | 79.7 | 3.6B | 531 ms |
| **GaussianDream Variants** | | | | | | | | | | | |
| GaussianDream-Sim† | – | 73.8 | 72.5 | 88.2 | 97.5 | 97.0 | 90.3 | 88.7 | 85.8 | 3.6B | 531 ms |
| **GaussianDream** | 98.4 | **77.3** | 72.1 | 86.9 | **98.9** | **98.5** | **93.6** | **88.4** | **87.0** | 3.6B | 531 ms |

*LingBot-VA reports 1220 ms on B200 and 5560 ms on A100; all other latency values are measured on A100. Only the LIBERO-Long checkpoint of LingBot-VA is publicly available. †GaussianDream-Sim uses privileged simulator-provided ground-truth depth and scene flow and is included only for analysis.

policy. The gains are most pronounced on tasks requiring precise object localization and placement, where current Gaussian reconstruction and future prediction jointly capture scene geometry and short-horizon evolution for precise manipulation. GeoPredict (Qian et al., 2025) performs better on Doors/Drawers and Others, where partial observability and out-of-view motion can limit our pixel-based reconstruction and prediction. Nevertheless, GaussianDream achieves the best overall performance and outperforms Being-H0.5 (Luo et al., 2026).

**Real robot.** As shown in Fig. 3, on the AgileX Piper platform, GaussianDream improves the average success rate from 34.4% to 50.0% over $\pi_{0.5}$ (Physical Intelligence et al., 2025), with improvements across all four task categories. The gains are particularly evident in spatial-relation reasoning and long-horizon goal planning. On the dual-UR7e platform, GaussianDream further achieves a 40.0% success rate on the higher-precision task, providing additional validation in a dual-arm setting.

### 4.3 Ablation and Analysis

| | | | | |
|---|---|---|---|---|
| Current Reconstruction | ✓ | ✓ | ✓ | ✓ |
| Future Prediction | ✗ | ✓ | ✓ | ✓ |
| Rendering Branch | ✓ | ✗ | ✓ | ✓ |
| Depth Branch | ✓ | ✓ | ✗ | ✓ |
| Avg. on LIBERO | 97.3 (-1.1) | 97.5 (-0.9) | 97.2 (-1.2) | **98.4** |
| Avg. on LIBERO-Plus | 86.0 (-1.0) | 86.2 (-0.8) | 85.9 (-1.1) | **87.0** |

Table 4: Component ablation of GaussianDream on LIBERO and LIBERO-Plus. Values in parentheses indicate the performance drop relative to the full model.

| Steps | PSNR ↑ | SSIM ↑ | EPE3D ↓ | Success ↑ |
|---|---|---|---|---|
| 42K | 18.04 | 0.680 | 0.0914 | 85.6 |
| 60K | 18.06 | 0.657 | 0.0883 | **87.0** |
| 80K | **19.92** | **0.722** | **0.0815** | 85.7 |

Table 5: Reconstruction and prediction quality versus policy performance. Better reconstruction metrics do not necessarily lead to higher policy success on LIBERO-Plus.

**Component analysis.** Compared with the base $\pi_0.5$ policy, introducing current Gaussian reconstruction improves the LIBERO-Plus success rate from 84.6% to 86.0%, showing that explicit scene reconstruction provides useful spatial supervision for policy learning. Building on this reconstruction objective, removing future prediction reduces the average success rate from 98.4% to 97.3% on LIBERO and from 87.0% to 86.0% on LIBERO-Plus. This indicates that current reconstruction and future prediction provide complementary supervision for policy learning.

**Supervision analysis.** As shown in Tab. 4, removing the rendering branch reduces the average success rate from 98.4% to 97.5% on LIBERO and from 87.0% to 86.2% on LIBERO-Plus. Removing the depth branch results in a larger drop to 97.2% and 85.9%, respectively. These results show that rendering and depth supervision are complementary, with depth supervision contributing more strongly, as removing it causes a larger performance drop.

**Reconstruction quality and policy performance.** As shown in Tab. 5, better reconstruction and prediction metrics do not always correspond to higher policy success. The 60K checkpoint achieves the best LIBERO-Plus success rate of 87.0%, whereas the 80K checkpoint obtains better PSNR, SSIM, and EPE3D but lower policy performance. A similar trend is observed in Tab. 3. GaussianDream-Sim replaces pseudo geometric supervision with simulator-provided ground-truth depth and scene flow. Although this privileged supervision improves performance over the base $\pi_{0.5}$ policy (85.8% vs. 84.6%), it does not outperform the final GaussianDream model (87.0%). This result further suggests that stronger or more accurate geometric supervision does not necessarily lead to better control performance. Together, these results indicate that the role of reconstruction and future prediction is not to maximize world-model fidelity in isolation, but to provide sufficiently informative spatial and temporal constraints for policy representation learning. Maintaining an appropriate balance between world modeling and action learning is therefore important for downstream control.

### 4.4 Qualitative Analysis

Fig. 4 shows that GaussianDream preserves strong spatial structure in both current reconstruction and future prediction, while also capturing short-horizon scene evolution. The interaction example further shows that GaussianDream can model object manipulation, such as the robot grasping and moving a bottle.

We further evaluate the spatial information encoded in learned visual representations through frozen-backbone depth probing. With the visual backbone frozen, the probe examines the geometric information already encoded in the learned features. We compare GaussianDream with representative methods from different paradigms, including $\pi_{0.5}$ (VLA baseline), Fast-WAM (world-action model baseline), RoboFlow4D (3D scene-flow-based method), and Spatial Forcing (3D-enhanced VLA). GaussianDream yields more accurate depth predictions, indicating that its visual representations preserve stronger geometric information and finer spatial structure. More rendering results are provided in the appendix.

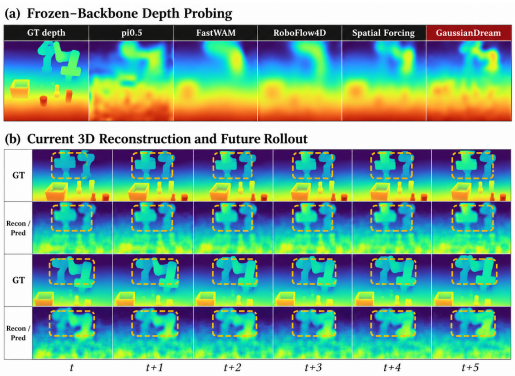


Figure 4: Qualitative analysis of GaussianDream. (a) Frozen-backbone depth probing across representative methods. (b) Current 3D Gaussian reconstruction and short-horizon future prediction.

## 5 Conclusion

We presented GaussianDream, a feed-forward 3D Gaussian world model for language-conditioned robotic manipulation. GaussianDream jointly learns current-scene reconstruction and short-horizon future prediction in a unified 3D Gaussian space, using RGB, pseudo depth, and pseudo 3D scene flow to provide structured spatial and dynamic supervision for VLA policy learning. Through a two-stage training strategy, the GaussianDream prefix is first pretrained with reconstruction and prediction objectives and then jointly optimized with the action policy. At inference time, all auxiliary Gaussian decoding branches are removed, and only the learned prefix is retained for action generation, requiring neither explicit Gaussian reconstruction nor future rollout.

Experiments on LIBERO, LIBERO-Plus, RoboCasa Human-50, and single- and dual-arm real robots demonstrate strong performance and cross-distribution generalization. Ablation studies and depth probing further validate the benefits of current reconstruction, future prediction, and geometric supervision, while showing that better reconstruction quality does not necessarily translate into higher policy success. By representing current geometry and future scene changes within the same Gaussian space, GaussianDream provides a shared spatial-temporal representation that directly supports downstream control for robust action generation. These results suggest that structured world-model supervision can enrich control-oriented representations without increasing deployment complexity. Overall, GaussianDream provides an effective framework for integrating 3D spatial modeling, future geometric prediction, and action learning.

## AI USE STATEMENT

Generative AI tools were used to assist with language editing and software debugging during the preparation of this work. All AI-assisted outputs were reviewed and verified by the authors.

## REFERENCES


Ali Abouzeid, Malak Mansour, Zezhou Sun, and Dezhen Song. GeoAware-VLA: Implicit geometry aware vision-language-action model. *CoRR*, abs/2509.14117, 2025. URL https://arxiv.org/abs/2509.14117.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *CoRR*, abs/2512.13030, 2025. URL https://arxiv.org/abs/2512.13030.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. RT-1: Robotics transformer for real-world control at scale. *CoRR*, abs/2212.06817, 2022. URL https://arxiv.org/abs/2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818, 2023. URL https://arxiv.org/abs/2307.15818.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Shengliang Deng, Mi Yan, Yixin Zheng, Jiayi Su, Wenhao Zhang, Xiaoguang Zhao, Heming Cui, Zhizheng Zhang, and He Wang. StereoVLA: Enhancing vision-language-action models with stereo vision. *CoRR*, abs/2512.21970, 2025. URL https://arxiv.org/abs/2512.21970.

Xianzhe Fan, Shengliang Deng, Xiaoyang Wu, Yuxiang Lu, Zhuoling Li, Mi Yan, Yujia Zhang, Zhizheng Zhang, He Wang, and Hengshuang Zhao. Any3d-vla: Enhancing vla robustness via diverse point clouds. *arXiv preprint arXiv:2602.00807*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *CoRR*, abs/2406.09246, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. *CoRR*, abs/2601.16163, 2026. URL https://arxiv.org/abs/2601.16163.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *CoRR*, abs/2510.12276, 2025a. URL https://arxiv.org/abs/2510.12276.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *CoRR*, abs/2601.21998, 2026a. URL https://arxiv.org/abs/2601.21998.

Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *Advances in Neural Information Processing Systems*, 38:63635–63673, 2026b.

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. QDepth-VLA: Quantized depth prediction as auxiliary supervision for vision-language-action models. *CoRR*, abs/2510.14836, 2025b. URL https://arxiv.org/abs/2510.14836.

Sixu Lin, Junliang Chen, Huaiyuan Xu, Zhuohao Li, Guangming Wang, Yixiong Jing, Sheng Xu, Runyi Zhao, Brian Sheil, Lap-Pui Chau, et al. Roboflow4d: A lightweight flow world model toward real-time flow-guided robotic manipulation. *arXiv preprint arXiv:2605.17522*, 2026. URL https://arxiv.org/abs/2605.17522.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: Towards scalable gaussian world models for robotic manipulation. In *ICCV*, pp. 9263–9274, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lu_GWM_Towards_Scalable_Gaussian_World_Models_for_Robotic_Manipulation_ICCV_2025_paper.html.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Chaojun Ni, Cheng Chen, Xiaofeng Wang, Zheng Zhu, Wenzhao Zheng, Boyuan Wang, Tianrun Chen, Guosheng Zhao, Haoyun Li, Zhehao Dong, Qiang Zhang, Yun Ye, Yang Wang, Guan Huang, and Wenjun Mei. SwiftVLA: Unlocking spatiotemporal dynamics for lightweight vla models at minimal overhead. *CoRR*, abs/2512.00903, 2025. URL https://arxiv.org/abs/2512.00903.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, et al. Octo: An open-source generalist robot policy. *CoRR*, abs/2405.12213, 2024. URL https://arxiv.org/abs/2405.12213.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *CoRR*, abs/2501.09747, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025. URL https://arxiv.org/abs/2504.16054.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. GeoPredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. *CoRR*, abs/2512.16811, 2025. URL https://arxiv.org/abs/2512.16811.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics. *CoRR*, abs/2506.01844, 2025. URL https://arxiv.org/abs/2506.01844.

Zijian Song, Qichang Li, Jiawei Zhou, Zhenlong Yuan, Tianshui Chen, Liang Lin, and Guangrun Wang. Robotic manipulation is vision-to-geometry mapping ($f(v) \rightarrow g$): Vision-geometry backbones over language and video models. *CoRR*, abs/2604.12908, 2026. URL https://arxiv.org/abs/2604.12908.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. GeoVLA: Empowering 3d representations in vision-language-action models. *CoRR*, abs/2508.09071, 2025. URL https://arxiv.org/abs/2508.09071.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fu-Cheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. Lamp: Learning vision-language-action policies with 3d scene flow as latent motion prior. *arXiv preprint arXiv:2603.25399*, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *CoRR*, abs/2409.12514, 2024. URL https://arxiv.org/abs/2409.12514.

Zhenyu Wu, Yuheng Zhou, Xiuwei Xu, Ziwei Wang, and Haibin Yan. Momanipvla: Transferring vision-language-action models for general mobile manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1714–1723, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Yuke Zhu, Linxi Fan, and Joel Jang. World Action Models are Zero-shot Policies. *CoRR*, abs/2602.15922, 2026. URL https://arxiv.org/abs/2602.15922.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *Advances in Neural Information Processing Systems*, 38:93409–93439, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

Wei Zhao, Pengxiang Ding, Zhang Min, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. In *International conference on learning representations*, volume 2025, pp. 51676–51693, 2025.

Hanyu Zhou, Chuanhao Ma, and Gim Hee Lee. VLA-4D: Embedding 4d awareness into vision-language-action models for spatiotemporally coherent robotic manipulation. *CoRR*, abs/2511.17199, 2025. URL https://arxiv.org/abs/2511.17199.

# APPENDIX

This appendix provides additional details and analyses of GaussianDream. Sec. A summarizes the notation used throughout the method. Sec. B describes the architecture and training configuration, including the GaussianDream token pathway, the Temporal Gaussian Evolution (TGE) Module, the Gaussian decoder, and the two-stage optimization procedure. Sec. C details the construction of pseudo depth and multi-horizon 3D scene-flow supervision, while Sec. D provides the complete reconstruction, rendering, scene-flow, and policy objectives. Sec. E presents the hardware and data-collection setups for the single-arm Piper and dual-UR7e experiments. Sec. F provides supplementary real-robot analyses of execution smoothness and inference efficiency. Finally, Sec. G presents additional qualitative visualizations of current reconstruction and short-horizon future prediction.

## A. NOTATION SUMMARY

| Symbol | Name | Meaning |
|---|---|---|
| $\mathbf{o}_t$, $\mathbf{o}_{t-K:t}$ | Observations | Current multi-view RGB observation and temporal observation window. |
| $\mathbf{l}$, $\mathbf{s}_t$, $\mathbf{a}_t$ | Language, state, action | Language instruction, robot state, and predicted action chunk. |
| $\pi_\theta$ | Action policy | Base continuous-action policy conditioned on the GaussianDream prefix. |
| $\mathbf{Z}_t^{\mathrm{GD}}$ | GaussianDream prefix | Learned prefix tokens used for current reconstruction, future prediction, and action conditioning. |
| $\mathbf{Q}_{\mathrm{GD}}$, $\mathcal{F}_\omega$ | GaussianDream queries/encoder | Learnable queries and temporal encoder used to build $\mathbf{Z}_t^{\mathrm{GD}}$ from multi-frame visual evidence. |
| TGE Module | Temporal Gaussian Evolution Module | Module that aggregates multi-frame 3D-aware visual tokens and GaussianDream queries. |
| $\mathcal{B}_{\mathrm{G}}$ | Gaussian decoder backbone | Decoder backbone that maps $\mathbf{Z}_t^{\mathrm{GD}}$ to dense Gaussian feature maps. |
| $\mathbf{F}_t^{\mathrm{G}}$ | Gaussian feature map | Shared dense feature map produced by the Gaussian decoder. |
| $\mathcal{R}_\phi$ | Reconstruction decoder | Auxiliary decoder for current Gaussian reconstruction. |
| $\mathcal{D}_\psi$ | Prediction decoder | Auxiliary decoder for future Gaussian prediction. |
| $\mathcal{G}_t$ | Current Gaussian state | Reconstructed current 3D Gaussian state. |
| $\hat{\mathcal{G}}_{t+\Delta}$ | Future Gaussian state | Predicted Gaussian state at future horizon $\Delta$. |
| $\mathbf{D}_t$, $\Theta_t$ | Depth / attributes | Current depth map and Gaussian attribute map. |
| $\Theta_t^{\mathrm{geo}}$, $\Theta_t^{\mathrm{app}}$ | Geometry / appearance attributes | Gaussian geometry attributes and spherical-harmonics appearance coefficients. |
| $\boldsymbol{\mu}_i^t$, $\theta_i^t$ | Gaussian primitive | Center and non-positional attributes of the $i$-th current Gaussian. |
| $\hat{\boldsymbol{\mu}}_i^{t+\Delta}$ | Future center | Predicted future center of the $i$-th Gaussian. |
| $\boldsymbol{\nu}_t^{(\Delta)}$, $\Delta\mathbf{X}_t^{(\Delta)}$ | Motion prediction | Horizon-conditioned velocity and center displacement. |
| $\mathbf{e}_\Delta$, $\alpha_\Delta$ | Horizon terms | Horizon embedding and time-scale factor. |

Table 6: Core notation for GaussianDream and its Gaussian reconstruction/prediction modules.

| Symbol | Name | Meaning |
|---|---|---|
| $\mathcal{H}$, $w_\Delta$ | Horizons | Future horizon set and horizon-specific loss weight. |
| $\mathcal{L}_{\mathrm{cur}}$ | Current loss | Current depth and rendering reconstruction loss. |
| $\mathcal{L}_{\mathrm{fut}}$ | Future loss | Future depth, rendering, and pseudo scene-flow supervision loss. |
| $\mathcal{L}_{\mathrm{GD}}$ | GaussianDream objective | Combined current reconstruction and future prediction objective. |
| $\mathbf{c}_t$ | Policy context | Action-conditioning context with visual observation, language, robot state, and GaussianDream prefix. |
| $\tau$, $\boldsymbol{\epsilon}$ | Flow variables | Flow-matching time and Gaussian noise. |
| $\mathbf{v}_\theta$ | Flow field | Policy-predicted velocity field in the action objective. |
| $\mathcal{L}_{\mathrm{act}}$ | Action loss | Flow-matching action loss from the base $\pi_{0.5}$ policy. |

Table 7: Training notation used in the GaussianDream and action objectives.

## B. Architecture Details

This section provides implementation details for GaussianDream described in Sec. 3. The architecture enables current 3D Gaussian reconstruction and short-horizon future prediction during training while keeping the online policy lightweight. The GaussianDream token path encodes temporal 3D-aware information into a compact prefix, which is decoded into current and future Gaussian states and supervised by RGB, depth, and pseudo 3D scene flow.

The GaussianDream token path uses a dense 32×32 query grid, corresponding to 1024 learnable GaussianDream queries. Each query is first projected to 512 channels and processed by the TGE Module. Each TGE Module block performs self-attention over the joint query–patch tokens within each frame, followed by temporal attention over the same token slot across frames. We use 8 attention heads and a 4× expansion MLP. The temporal visual tokens are obtained from VGGT (Wang et al., 2025) feature taps using 32×32 adaptive average pooling and linear projection to 512 channels. The TGE Module output at the latest frame is projected back to the VLM width and appended to the multimodal prefix as the GaussianDream token segment. After VLM prefix processing, this segment becomes $\mathbf{Z}_t^{\mathrm{GD}}$.

The Gaussian decoder reshapes $\mathbf{Z}_t^{\mathrm{GD}}$ into a canonical 32×32 token grid and applies a shared token-to-feature backbone. The backbone uses three transposed-convolution upsampling blocks with kernel size 4, stride 2, and padding 1, progressively lifting the token grid to a 256×256 feature map while reducing channels from the VLM width to 512, 256, and finally 128. Each upsampling block contains GroupNorm, GELU, a 3×3 convolution, and a bilinear residual skip. The resulting 128-channel feature map is further refined by DPT-style feature fusion blocks with 3×3 and 1×1 convolutions and residual connections from intermediate resolutions.

The current reconstruction head operates on the shared 256×256 feature map. Its geometry branch predicts 8 channels with a 3×3 convolution: quaternion rotation (4), scale (3), and opacity (1). A separate depth branch uses two 3×3 convolutional layers with GroupNorm and GELU, followed by a final 3×3 convolution to predict one depth channel. The appearance branch fuses the RGB image through a $7 \times 7$ convolution and predicts 9 directional degree-1 spherical-harmonic channels with a $3 \times 3$ convolution, while the RGB image provides the 3 base-color (DC) channels. Depth is unprojected with camera intrinsics to form Gaussian centers, while the remaining channels provide Gaussian attributes.

The future prediction head reuses the same shared 256×256 feature map. A horizon embedding is added after a 3×3 projection to 128 channels, followed by a residual block with two 3×3 convolutions and GroupNorm. The velocity output head then applies a 3×3 convolution, GroupNorm, GELU, and a final 1×1 convolution to predict a 3-channel center-velocity map. The velocity is passed through `tanh` and scaled, then multiplied by the horizon time factor to update only Gaussian centers; scale, opacity, appearance, and rotation are copied from the current Gaussian template. This design concentrates future prediction on short-horizon geometric change while preserving the stability of the reconstructed current Gaussian state.

**Training and Evaluation Details.** GaussianDream is trained in two stages for a total of 60K optimization steps. In Stage I, we pretrain the GaussianDream representation for 20K steps using only the current-scene reconstruction and future prediction objectives. In Stage II, we initialize from the Stage-I checkpoint and jointly optimize the GaussianDream representation and the action policy for an additional 40K steps. The GaussianDream objectives remain active during Stage II. Detailed loss weights are provided in Sec. D.

We use a global batch size of 24 and AdamW as the optimizer, with a peak learning rate of $5 \times 10^{-5}$ and a cosine learning-rate schedule with 10K warmup steps. Gradient clipping is applied with a maximum norm of 1.0, and we maintain an exponential moving average (EMA) of model parameters with a decay rate of 0.999. All experiments are conducted with mixed-precision training on NVIDIA A100 GPUs.

Unless otherwise specified, step counts refer to the total number of optimization steps across both stages. The 60K checkpoint is used as the default final model, while the 80K checkpoint is obtained by extending Stage-II training for the reconstruction–control analysis.

During evaluation on LIBERO (Liu et al., 2023) and RoboCasa (Nasiriany et al., 2024), each task is executed for 50 trials following the corresponding benchmark protocols. The reported success rates are averaged across all evaluation trials.

## C. Pseudo Depth and Scene-Flow Preparation

GaussianDream uses pseudo geometric supervision to train current Gaussian reconstruction and future Gaussian prediction. For each demonstration frame, we first generate a dense pseudo depth map from the agent-view RGB observation using the metric-depth variant of Depth Anything V2 (Yang et al., 2024). The predicted depth map is resized to the original image resolution and stored together with the corresponding episode data. These pseudo geometric targets are used only during training and provide additional spatial and motion supervision beyond the standard action-prediction objective.

To construct motion supervision, we first pre-compute pairwise pseudo 3D scene-flow sidecars from temporally adjacent observation pairs. Given two consecutive RGB observations $(\mathbf{o}_t, \mathbf{o}_{t+1})$, we estimate a dense 2D optical flow field $\mathbf{f}_{t\to t+1}$ using RAFT (Teed & Deng, 2020) by default, with Farneback flow as a lightweight fallback. For each pixel $(u, v)$ in frame $t$, the corresponding image location in frame $t{+}1$ is

$$(u', v') = (u, v) + \mathbf{f}_{t\to t+1}(u, v). \tag{13}$$

We then bilinearly sample the future depth map at $(u', v')$ and back-project the corresponding pixels in the two frames using the camera intrinsics $\mathbf{K}$:

$$\mathbf{x}_t = \Pi^{-1}\big(u, v, \mathbf{D}_t(u, v); \mathbf{K}\big), \tag{14}$$

$$\mathbf{x}_{t+1} = \Pi^{-1}\big(u', v', \mathbf{D}_{t+1}(u', v'); \mathbf{K}\big). \tag{15}$$

The pairwise pseudo 3D scene-flow target is defined as

$$\mathbf{F}^{3D}_{t\to t+1}(u, v) = \mathbf{x}_{t+1} - \mathbf{x}_t. \tag{16}$$

For multi-horizon supervision, we do not directly estimate optical flow from $t$ to $t + \Delta$. Instead, during training, the pairwise sidecars are composed along the tracked 2D correspondence trajectory. Starting from an anchor pixel in frame $t$, we iteratively sample the pairwise 2D and 3D flow at the current tracked location, update the 2D track, and accumulate the corresponding 3D displacements. This produces an anchor-aligned pseudo 3D scene-flow target $\mathbf{F}^{3D,(\Delta)}_{t\to t+\Delta}$ for each prediction horizon. In our LIBERO configuration, scene-flow supervision is applied to $\Delta \in \{1, 2, 3, 4, 5\}$.

We additionally compute a validity mask to exclude correspondences that are warped outside the image, pixels with invalid depth, and pixels outside the configured depth range. Invalid flow vectors are set to zero and ignored by the scene-flow supervision loss. Each pre-computed sidecar stores the pairwise 2D optical flow, pseudo 3D scene flow, validity mask, camera intrinsics, camera identifier, and episode/task metadata. These sidecars are loaded only during training and are not required at inference time.

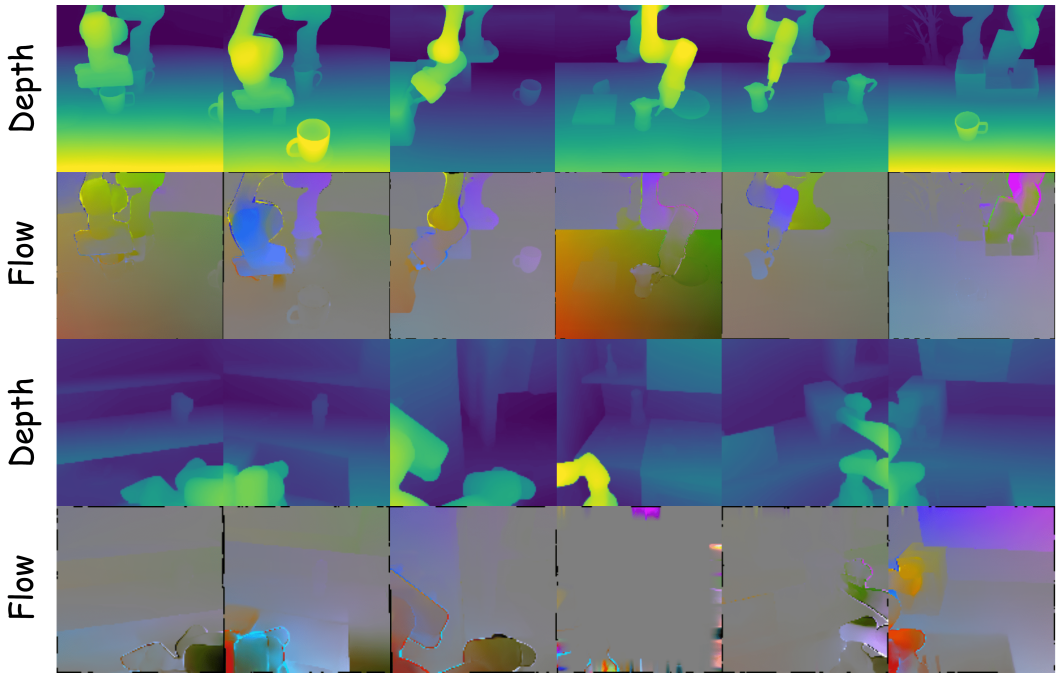


Figure 5: Visualization of pseudo depth and scene-flow preparation on LIBERO (Liu et al., 2023) and RoboCasa (Nasiriany et al., 2024) demonstrations. The first two rows show LIBERO depth maps and flow fields, while the last two rows show RoboCasa depth maps and flow fields. These visualizations illustrate the pairwise pseudo geometric supervision used to construct multi-horizon 3D scene-flow targets for future Gaussian prediction.

## D. Loss Details

We provide implementation details for the auxiliary objectives introduced in Sec. 3.4. The overall GaussianDream and policy objectives follow Eqs. (10)–(12) in the main paper.

For current-scene supervision, we use an $\ell_1$ depth loss between the rendered Gaussian depth and the pseudo or simulator-provided depth target. The rendering objective $\mathcal{L}_{\text{cur}}^{\text{render}}$ consists of an RGB photometric term, Gaussian scale/opacity regularization, and an edge-aware depth smoothness term.

The RGB photometric component is defined as

$$\mathcal{L}_{\text{rgb}} = 0.85\,\mathcal{L}_{\text{SSIM}} + 0.15\,\mathcal{L}_1 + 0.1\,\mathcal{L}_{\text{LPIPS}}. \tag{17}$$

For future prediction, the same depth and rendering objectives are applied at each prediction horizon $\Delta \in \mathcal{H}$. The future RGB photometric component additionally uses a motion-aware weighting map derived from the current and future RGB/depth observations. The predicted Gaussian center displacement is further supervised by pseudo 3D scene flow. With a validity mask, the flow loss is

$$\mathcal{L}_{\text{flow}}^{(\Delta)} = \frac{\sum_i \mathbf{M}_i^{(\Delta)} \left\| \Delta\mathbf{x}_i^{(\Delta)} - \mathbf{F}_i^{3D,(\Delta)} \right\|_1}{\sum_i \mathbf{M}_i^{(\Delta)} + \epsilon}. \tag{18}$$

Here, $\mathbf{M}_i^{(\Delta)}$ denotes the validity mask and $\mathbf{F}_i^{3D,(\Delta)}$ denotes the anchor-aligned pseudo 3D scene-flow target at the corresponding Gaussian or pixel location. Invalid correspondences are excluded from the loss.

Unless otherwise specified, we use $\lambda_{\text{cur}}^{\text{depth}} = \lambda_{\text{depth}} = 0.1$, $\lambda_{\text{cur}}^{\text{render}} = \lambda_{\text{render}} = 0.2$, and $\lambda_{\text{flow}} = 15.0$, with uniform horizon weights $w_\Delta = 1.0$ for $\Delta \in \{1, 2, 3, 4, 5\}$. During Stage II, the GaussianDream objective is retained with $\lambda_{\text{GD}} = 0.1$.

All reconstruction and prediction losses are used only during training. At inference time, the Gaussian decoding heads are removed and the GaussianDream prefix is retained for policy conditioning.




Figure 6: Real-robot hardware setup. Left: single-arm leader–follower teleoperation setup, where a human operates the leader arm and the corresponding motions are transferred to the follower arm used for policy learning and evaluation. The follower platform is equipped with an agent-view RGB camera and a wrist-mounted RGB camera. Right: dual-UR7e platform with VR-based teleoperation for demonstration collection. During autonomous evaluation, GaussianDream directly controls the corresponding robot platform from language instructions, robot states, and visual observations.

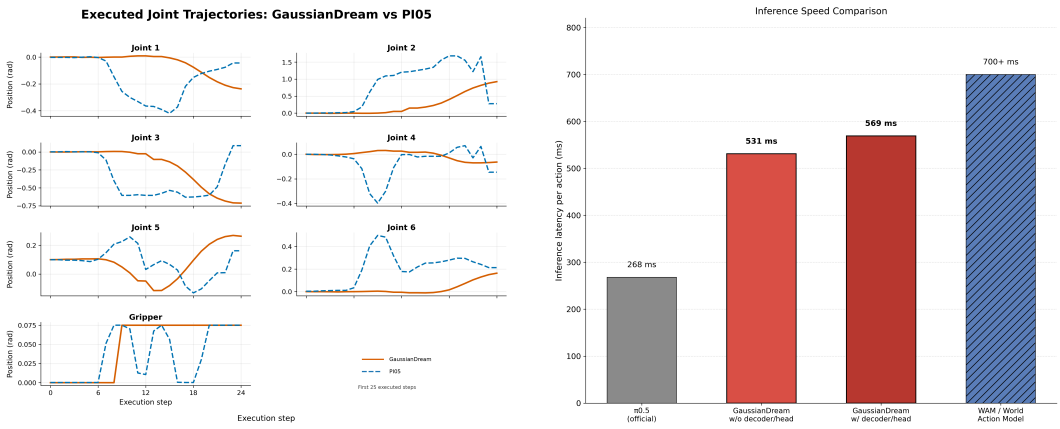


Figure 7: Real-robot smoothness and inference-speed analysis. Left: execution smoothness comparison between GaussianDream and the $\pi_{0.5}$ (Physical Intelligence et al., 2025) baseline. Right: per-action-chunk inference latency comparison across GaussianDream variants and representative WAM baselines; detailed baseline results are reported in the main paper.

## E. Real-Robot Hardware Setup

We evaluate GaussianDream on both single-arm and dual-arm robotic platforms. For the single-arm experiments, we use a leader–follower teleoperation setup. A human operator controls the leader arm, and the corresponding motions are transferred to the policy-controlled follower arm. The platform is equipped with an agent-view RGB camera and a wrist-mounted RGB camera for visual observation. For each Piper task, we collect 100 demonstrations and evaluate the learned policy over 20 trials.

For the dual-arm experiments, we use a dual-UR7e platform and collect demonstrations through VR-based teleoperation. During autonomous evaluation, GaussianDream controls the robot from language instructions, robot states, and visual observations.

At inference time, three temporal observations $\{t-10, t-5, t\}$ are encoded into GaussianDream prefix tokens and injected into the policy context. The auxiliary Gaussian reconstruction and future-prediction heads are removed during deployment.

## F. Real-Robot Supplementary Results

We provide additional real-robot analysis to complement the physical success rates reported in the main paper. As shown in Fig. 7, GaussianDream exhibits smoother execution trajectories than the $\pi_{0.5}$ (Physical Intelligence et al., 2025) baseline, with fewer abrupt changes during manipulation. The same figure also illustrates the efficiency of the asymmetric training–inference design. After

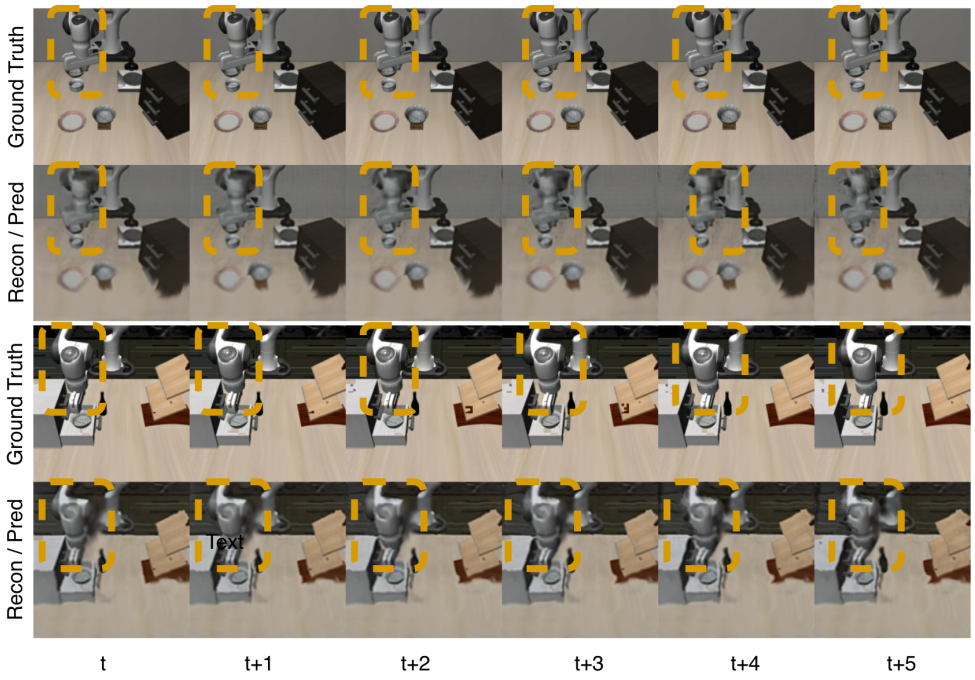


Figure 8: Additional RGB rendering results of current Gaussian reconstruction and future Gaussian prediction. Each example compares the observed frame with the rendered current scene and predicted future renderings. Additional depth-based analysis is provided in the main paper.

removing the Gaussian reconstruction and prediction heads at deployment, GaussianDream requires 531 ms per chunk, compared with 569 ms for the diagnostic configuration that retains these auxiliary heads. Both configurations remain faster than representative WAM baselines, which require more than 700 ms per action chunk; detailed latency comparisons are reported in the main paper.

## G. Additional Qualitative Visualization

We provide additional qualitative results of RGB renderings from current Gaussian reconstruction and short-horizon future prediction. The current reconstruction results preserve the appearance and spatial arrangement of the observed scene, while the future renderings capture visual changes in subsequent observation frames. These results complement the depth-based spatial analysis presented in the main paper.