Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

GS-Playground: A High-Throughput Photorealistic Simulator for Vision-Informed Robot Learning

Yufei Jia[1*], Heng Zhang[2*], Ziheng Zhang[3*], Junzhe Wu[1*], Mingrui Yu[1*], Zifan Wang[5], Dixuan Jiang[6], Zheng Li[5], Chenyu Cao[7], Zhuoyuan Yu[3], Xun Yang[5], Haizhou Ge[1], Yuchi Zhang[4], Jiayuan Zhang[4], Zhenbiao Huang[8], Tianle Liu[3], Shenyu Chen[9], Jiacheng Wang[3], Bin Xie[3], Xuran Yao[4], Xiwa Deng[2], Guangyu Wang[1], Jinzhi Zhang[1], Lei Hao[10], Zhixing Chen[1], Yuxiang Chen[11], Anqi Wang[4], Hongyun Tian[3], Yiyi Yan[4], Zhanxiang Cao[12], Yizhou Jiang[1], Hanyang Shao[4], Yue Li[4], Lu Shi[1], Wei Sui[13], Hanqing Cui[2], Yusen Qin[13], Ruqi Huang[1†], Bokui Chen[1†], Lei Han[4†], Tiancai Wang[3†], Guyue Zhou[1†]

[1]THU, [2]Motphys, [3]Dexmal, [4]DISCOVER Robotics, [5]HKUST(GZ), [6]BIT, [7]Galbot, [8]NUS, [9]HITSZ, [10]XJTU, [11]NJU, [12]SJTU, [13]D-Robotics

[*]Equal contribution. [†]Advising. Correspondence to: Yufei Jia <jyf23@mails.tsinghua.edu.cn>.

Fig. 1: **`GS-Playground` Overview.** It integrates photorealistic 3D Gaussian Splatting with high-performance parallel physics, achieving over 10^4 aggregate FPS across 2048 environments at 640×480 resolution on a single GPU. We provide comprehensive sensor suites (Contact, Vision, LiDAR) and support a wide range of robotic embodiments and learning tasks, including locomotion, navigation, and manipulation.

***Abstract*—Embodied AI research is undergoing a shift toward vision-centric perceptual paradigms. While massively parallel simulators have catalyzed breakthroughs in proprioception-based locomotion, their potential remains largely untapped for vision-informed tasks due to the prohibitive computational overhead of large-scale photorealistic rendering. Furthermore, the creation of simulation-ready 3D assets heavily relies on labor-intensive manual modeling, while the significant sim-to-real physical gap hinders the transfer of contact-rich manipulation policies. To address these bottlenecks, we propose `GS-Playground`, a multi-modal simulation framework for high-throughput, photorealistic robot learning in interactive rigid-body settings. We develop MotrixSim, a novel high-performance parallel physics engine, specifically designed to integrate with a batch 3D Gaussian Splatting (3DGS) rendering pipeline to ensure high-fidelity synchronization. Our system achieves an aggregate rendering throughput of 10^4 FPS across 2048 environments at 640×480 resolution, significantly lowering the barrier for large-scale visual RL. Additionally, we introduce a Real2Sim workflow that reconstructs photorealistic and memory-efficient simulation-ready environments, reducing the manual effort required to create complex scenes for robot interaction. Extensive experiments on locomotion, navigation, and manipulation demonstrate that `GS-Playground` enables high-throughput photorealistic RL and robust Sim2Real transfer for interactive rigid-body robot learning. Project homepage: https://gsplayground.github.io.**

I. Introduction

Vision serves as the most information-rich modality for robotic perception of the environment. Recently, significant progress has been made in learning policies for quasi-static

TABLE I: **Comparison of physical and perceptual capabilities across parallel robotics simulators.**

Simulators	Physics Engine	Batch Physics	VRAM Usage	Integrated Batch IK	Batch Renderer	Batch Render Fidelity	3DGS Env. Num.	Dynamic 3DGS Scene	3DGS Render FPS	Startup Speed	Physics Cross Platform
MuJoCo/MJX [46]	Brax/MJX	CPU/GPU	⋆⋆	×	Madrona	+	-	-	-	+	L
IsaacLab [39]	PhysX5	GPU	⋆⋆⋆⋆⋆	✓	omni.RTX	++	-	-	-	++	L
ManiSkill [45]	PhysX5	GPU	⋆⋆⋆	✓	Vulkan SBR	+	-	-	-	+++	W/L
Genesis [64]	Taichi	GPU	⋆⋆	✓	Madrona	+	-	-	-	++	W/L/M
RoboGSim [29]	PhysX4	-	-	×	-	+++	1	✓	~ 100	++	L
DISCOVERSE [19]	MuJoCo	-	-	×	-	+++	$1 \sim 4$	✓	~ 650	++	L
GSWorld [20]	PhysX5	-	-	×	-	+++	1	✓	-	++	L
GaussGym [11]	PhysX4	GPU	⋆	×	GSplat	+++	Up To 4096	×	-	++	L
`GS-Playground`	MotrixSim	CPU/GPU	-	✓	BatchSplat	+++	Up To 4096	✓	$\sim$ **10k**	++++	W/L/M

Note: 1. **Batch Physics**: Indicates supported hardware for batched simulation. 2. **VRAM Usage**: The number of '⋆' indicates higher VRAM consumption. In headless mode, only physics simulation overhead is considered. 3. **Batch Render Fidelity**: The number of '+' represents higher visual fidelity. 4. **3DGS Render FPS**: Aggregate throughput across 2048 environments at 640×480 resolution with an NVIDIA RTX 4090 GPU and an Intel i9-14900K CPU. 5. **Startup Speed**: More '+' means faster startup times. 6. **Physics Cross Platform**: W: Windows, L: Linux, M: macOS.

manipulation and navigation directly from real-world visual data [25, 65, 4, 59, 28, 53, 17]. However, tasks involving complex dynamics and contacts—such as locomotion and contact-rich manipulation—require large-scale parallel simulation for effective reinforcement learning (RL). While current massively parallel simulators have revolutionized proprioception-based learning [34, 39, 58, 45, 64], they often struggle to reconcile visual fidelity with rendering efficiency, limiting the application of large-scale, vision-informed policy learning. We attribute this gap to two primary limitations:

- **Prohibitive Rendering Overhead**: Existing simulation pipelines face severe scalability bottlenecks when integrating high-resolution rendering. The exorbitant computational cost of photorealistic rendering creates intense resource contention with policy learning, frequently resulting in Out-of-Memory (OOM) failures. Consequently, these hardware constraints force a compromise between visual fidelity and simulation throughput, restricting the scale and efficacy of vision-based training. Recent 3DGS-based robotic systems such as GSWorld [20] and RoboGSim [29] attach photorealistic rendering to manipulation pipelines, but run as single-environment closed-loop evaluators rather than batch-rendered, on-policy training engines.
- **Laborious Asset Synthesis**: Constructing simulation assets that combine visual fidelity with robot-interaction compatibility remains a persistent challenge. While 3D reconstruction has advanced significantly, converting these representations into "sim-ready" assets compatible with high-frequency rigid-body physics and memory-efficient rendering remains difficult and laborious. Vid2Sim [52] and Video2Game [51] reconstruct navigable scenes or playable games from a single video, but target agent navigation and graphics rather than robot-object interaction. This highlights the critical need for a pipeline capable of rapidly transforming individual real-world scenes into high-fidelity simulation environments for robot interaction.

To bridge the gap, we introduce `GS-Playground`, a robot-learning simulation framework that harmonizes high-throughput rigid-body physics simulation and high-fidelity visual rendering (Figure 1). Our platform maintains the precision and stability required for physics simulation while providing the rendering efficiency necessary for large-scale, vision-informed policy training and sim-to-real transfer. Our core contributions are as follows:

1) **Embodied Robot Simulation Platform:** We develop MotrixSim, a ground-up, cross-platform (Windows, Linux, and macOS) parallel physics engine that supports both GPU and CPU backends. This architecture provides high-fidelity rigid-body dynamics and comprehensive sensor integration (including RGB cameras, LiDAR, and force/contact sensors) across diverse robot embodiments (such as quadrupeds, humanoids, and manipulators). This platform facilitates a flexible development workflow from local prototyping to massively parallel training.
2) **Memory-Efficient Batch 3DGS Rendering:** To mitigate the memory bottlenecks in large-scale photorealistic simulation, we introduce a specialized point-pruning strategy optimized for rigid-body environments. This approach enables a memory-efficient rendering architecture capable of a breakthrough aggregate throughput of $\mathbf{10^4}$ **FPS** across 2048 environments at $\mathbf{640 \times 480}$ resolution on a single GPU, significantly expanding the scale of vision-based reinforcement learning.
3) **"Sim-Ready" Real2Sim Workflow:** We present a targeted pipeline that streamlines the conversion of individual real-world scenes into simulation-ready rigid-body environments with photorealistic rendering and task-configurable physical parameters. Our workflow significantly reduces the laborious manual effort usually required to create complex "sim-ready" assets, enabling the rapid population of diverse simulation environments.

We validate `GS-Playground` through a rigorous experimental regime. First, we demonstrate that our physics engine matches state-of-the-art simulators in accuracy and efficiency, with enhanced stability in specific contact-rich scenarios. Next, we confirm our rendering pipeline's ability to provide high-fidelity feedback at unprecedented speeds. Finally, we benchmark the framework on a diverse suite of tasks—including quadrupedal locomotion, humanoid control, navigation, and robotic manipulation—across both state-based

and vision-driven reinforcement learning. We summarize the key features of GS-Playground alongside other state-of-the-art simulators in Table I. We will release the framework and synthesized *Bridge-GS* dataset to empower the research community.

II. Related Work

A. Massive Parallelism in Simulation

Massive parallel physical simulation has emerged as the indispensable infrastructure for efficient robot learning, particularly for locomotion and contact-rich manipulation [18, 27, 43, 35, 10, 67, 36, 49, 16, 50, 63, 6]. Milestone platforms such as Isaac Gym [34], Isaac Lab [39], and Genesis [64] have revolutionized the throughput of sample collection by instantiating tens of thousands of parallel environments on GPUs. However, existing frameworks typically prioritize physical throughput over perceptual fidelity. Rendering architectures often diverge into two extremes: they either favor high sampling rates via streamlined rasterization (e.g., Madrona [42], ManiSkill3 [45]) or prioritize cinematic photorealism via computationally expensive ray-tracing (e.g., Isaac Lab [39]). The simultaneous achievement of massive parallel throughput with high-fidelity, photorealistic rendering remains a critical bottleneck for scaling vision-centric robot learning.

B. Vision-Centric Robot Learning

Vision serves as the most information-rich modality for robotic perception. Recently, significant progress has been made in learning policies for quasi-static manipulation and navigation directly from real-world visual data, exemplified by Vision-Language-Action (VLA) models [25, 65, 4] and Vision-Language-Navigation (VLN) models [9, 5, 60, 61]. However, tasks involving complex dynamics and intermittent contacts rely heavily on simulation-based reinforcement learning (RL) to acquire skills in an unsupervised manner.

Early attempts to incorporate visual inputs into RL were often constrained by conventional, small-scale simulations [2, 37, 66], where low simulation throughput hindered the stable acquisition of complex skills. While recent advancements in massive parallel simulation have enabled sophisticated policy optimization for locomotion and dexterous manipulation, these frameworks primarily rely on proprioceptive states or point clouds due to the prohibitive computational overhead and limited fidelity of visual rendering [7, 30, 23, 56]. Furthermore, to mitigate the sim-to-real gap, recent attempts on vision-based RL methods often necessitate extensive visual randomization of textures, lighting, and backgrounds [17, 53]. This, however, imposes a substantial burden on GPU resources, significantly raising the computational threshold for research in vision-informed robot learning.

C. Gaussian Splatting and Real-to-Sim Reconstruction

Building simulators highly consistent with the real world relies on high-quality visual rendering and precise physical modeling. Recently, Gaussian Splatting (3DGS) [24] has rapidly developed as an emerging scene reconstruction method, enabling photorealistic real-time rendering from arbitrary viewpoints. Its unique representation strikes a balance between visual fidelity and memory efficiency, making it particularly suitable for resource-constrained simulation environments. Recent research has significantly improved the capability to reconstruct renderable models from real scenes in terms of reconstruction paradigms [48, 38, 22], model efficiency [15, 12, 14], and generative Gaussian models [8, 31].

However, these methods generally struggle to directly meet the requirements of robotic simulators for sim-ready scenes. Existing works indicate that 3DGS-based rendering holds significant potential in robotics, enhancing sim-to-real transfer for vision-based policies [29, 13, 40], augmenting training datasets [3, 33, 57, 54], and supporting real-to-sim evaluation pipelines [21, 1, 20, 62]. While GaussGym [11] pioneered the application of 3DGS in RL, our work extends this capability to contact-rich manipulation and larger-scale parallel throughput, providing a more versatile foundation for diverse vision-informed robot learning tasks.

III. System Design

A. System Overview

Figure 2 illustrates the architecture of GS-Playground, which comprises three core tiers: (1) MotrixSim, a **high-performance parallel physics engine** supporting both GPU and CPU backends; (2) a **memory-efficient batch 3DGS renderer** optimized via point-pruning; and (3) an **automated Real2Sim pipeline** for rapid scene synthesis. These modules are seamlessly integrated with a multi-modal sensor suite to provide the high-throughput, photorealistic feedback necessary for vision-centric robot learning.

Data Flow. The simulation loop begins with the physics engine, which advances the world state using a velocity-impulse formulation. The updated rigid-body poses are synchronized with the Batch 3DGS Renderer through *Rigid-Link Gaussian Kinematics* (RLGK), enabling zero-overhead updates of visual clusters. The renderer produces photorealistic RGB images and depth maps, while the sensor suite provides LiDAR point clouds and high-dimensional contact data (including multi-point forces and torques). These modalities form the observation vector for end-to-end policy learning.

Asset Creation. Orthogonal to the loop, the *Real2Sim Pipeline* streamlines the conversion of individual real-world captures into simulation-ready assets. By performing automated segmentation, reconstruction, and sub-millimeter pose alignment, this pipeline produces photorealistic rigid-body assets for robot interaction.

Development Workflow. A key design principle is the flexibility of the development-to-training workflow. Our core physics engine is cross-platform (Windows, Linux, macOS), facilitating rapid local debugging and prototyping on various workstations. For large-scale training, the system leverages an optimized CUDA-based rendering pipeline on Linux, achieving a breakthrough aggregate throughput of 10^4 FPS across 2048 environments at 640×480 resolution by utilizing spe-

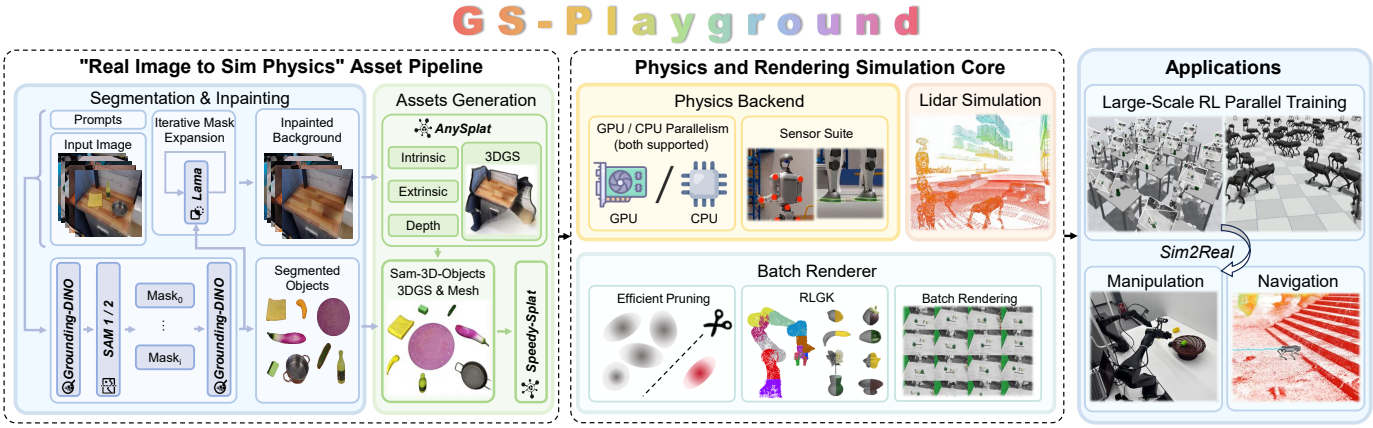


Fig. 2: **GS-Playground System Architecture. Left:** an automated Real2Sim pipeline that constructs simulation-ready visual and geometric assets from RGB inputs via object segmentation, background inpainting, and 3DGS/mesh reconstruction. **Middle:** a physics and rendering simulation core with CPU/GPU physics backends, integrated sensor and LiDAR simulation, and batch-optimized 3DGS rendering with pruning and rigid-link kinematics. **Right:** downstream applications including manipulation, navigation, and large-scale parallel reinforcement learning.

cialized point-pruning to balance visual fidelity and memory consumption.

B. Physics Solver Formulation

In robotic manipulation, the choice of constraint formulation directly impacts simulation fidelity. Optimization-centric solvers that rely on regularized soft contacts tend to produce visually smooth but physically 'spongy' interactions, where heavy payloads may exhibit gradual drift due to residual forces. MotrixSim utilizes a velocity-impulse formulation in generalized coordinates and implements strict complementarity with explicit velocity clamping at the friction limits. This approach sacrifices the smoothness of the gradients in exchange for geometric precision, allowing for the simulation of rigid bodies that can maintain perfect static equilibrium and allows for high constraint stiffness and large simulation time steps. This makes the engine particularly suitable for engineering applications where stability and exact constraint satisfaction are paramount.

The discretized dynamics equation for generalized coordinates $\mathbf{q} \in \mathbb{R}^n$ and velocities $\mathbf{v} \in \mathbb{R}^n$ over a time step h is formulated as:

$$\mathbf{M}(\mathbf{v}^+ - \mathbf{v}) = \mathbf{J}_e^T \boldsymbol{\lambda}_e^+ + \mathbf{J}_n^T \boldsymbol{\lambda}_n^+ + h(\boldsymbol{\tau}_{ext} - \mathbf{c}) \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{J}_e$ and $\mathbf{J}_n$ are the Jacobians for equality and inequality constraints, respectively, and $\boldsymbol{\lambda}$ denotes the constraint impulses. The term $\mathbf{c}$ accounts for Coriolis and centrifugal forces. We incorporate soft constraints by defining an implicit impulse relation $\boldsymbol{\lambda}^+ = f(\mathbf{u}^+; \mathbf{x}, h)$ and linearizing it via a first-order Taylor expansion at the current velocity $\mathbf{u}$:

$$\boldsymbol{\lambda}^+ \approx f(\mathbf{u}) + \frac{\partial f}{\partial \mathbf{u}}(\mathbf{u}^+ - \mathbf{u}) \tag{2}$$

By defining the positive definite compliance matrix $\mathbf{C} = (-\frac{\partial f}{\partial \mathbf{u}})^{-1}$ and the bias term $\boldsymbol{\zeta} = \mathbf{u} + \mathbf{C}f(\mathbf{u})$, we obtain the standardized compliance form:

$$\mathbf{u}^+ = -\mathbf{C}\boldsymbol{\lambda}^+ + \boldsymbol{\zeta} \tag{3}$$

By substituting this velocity relation into the constraint space and eliminating the equality constraints $\boldsymbol{\lambda}_e$ via the Schur complement method, we obtain a reduced linear system for the inequality constraints $\boldsymbol{\lambda}_n$:

$$\mathbf{u}_n^+ = \mathbf{A}\boldsymbol{\lambda}_n^+ + \mathbf{b} \tag{4}$$

The system matrix $\mathbf{A}$ and the right-hand side vector $\mathbf{b}$ are explicitly given by:

$$\mathbf{A} = \mathbf{J}_n \mathbf{M}^{-1} \mathbf{J}_n^T - \mathbf{J}_n \mathbf{M}^{-1} \mathbf{J}_e^T (\mathbf{W}_{ee} + \mathbf{C}_e)^{-1} \mathbf{J}_e \mathbf{M}^{-1} \mathbf{J}_n^T \tag{5}$$

$$\mathbf{b} = \mathbf{J}_n \tilde{\mathbf{v}} + \mathbf{J}_n \mathbf{M}^{-1} \mathbf{J}_e^T (\mathbf{W}_{ee} + \mathbf{C}_e)^{-1} (\boldsymbol{\zeta}_e - \mathbf{J}_e \tilde{\mathbf{v}}) \tag{6}$$

where $\mathbf{W}_{ee} = \mathbf{J}_e \mathbf{M}^{-1} \mathbf{J}_e^T$ is the effective inverse mass matrix of the equality constraints. The term $(\mathbf{W}_{ee} + \mathbf{C}_e)$ is guaranteed to be invertible as it is the sum of a positive semi-definite matrix and a positive definite matrix.

The solver resolves contact and friction as a Mixed Complementarity Problem (MCP). The impulse vector $\boldsymbol{\lambda}_n$ comprises normal components $\boldsymbol{\lambda}_\perp$ and frictional components $\boldsymbol{\lambda}_\parallel$. The solution must satisfy the bounds defined by the Coulomb friction model:

$$\begin{cases} w_i \geq 0, & \text{if } \lambda_i^+ = l_i \\ w_i = 0, & \text{if } l_i < \lambda_i^+ < u_i \\ w_i \leq 0, & \text{if } \lambda_i^+ = u_i \end{cases} \tag{7}$$

where $w_i = [(\mathbf{A}+\mathbf{C}_n)\boldsymbol{\lambda}_n^+ + (\mathbf{b} - \boldsymbol{\zeta}_n)]_i$. For normal contact, the bounds are $[0, \infty)$; for friction, the bounds are $[-\mu\lambda_\perp^+, \mu\lambda_\perp^+]$. This formulation is solved efficiently using a Projected Gauss-Seidel (PGS) solver, ensuring stable friction behavior while accommodating both rigid and compliant contact interactions.

This framework demonstrates high extensibility. It supports various physical constraints, including MJCF-defined contact models (e.g., parameters `solref`, `solimp`), tendons, and actuators. Integrating a new constraint type simply requires defining its impulse-state relationship $\boldsymbol{\lambda}(\mathbf{x}, \mathbf{u})$ and the corresponding Jacobian $\mathbf{J}$. Additionally, to achieve real-time

performance in large-scale scenarios with massive contacts, we implemented two key engineering optimizations:

1) **Parallelization via Constraint Islands:** Leveraging the spatial locality of physical interactions, we dynamically construct a constraint dependency graph at each time step. By analyzing the connectivity of the graph, the rigid body system is partitioned into disjoint sets of interacting bodies, termed "Constraint Islands." Since the Linear Complementarity Problems (LCPs) for these islands are mathematically independent, they are dispatched to multi-core CPU threads for parallel solving, ensuring linear performance scaling with scene complexity.

2) **Warm-Starting with Temporal Coherence:** We exploit the temporal coherence of physical processes by implementing a Contact Manifold Tracking system. This system persists contact constraints across simulation frames. Instead of initializing the Projected Gauss-Seidel (PGS) solver with zero vectors, we warm-start the solver using the converged impulses λ_{t-1} from the previous frame as the initial guess λ_{initial}. This strategy significantly accelerates convergence, typically reducing the required PGS iterations from over 50 to fewer than 10 for stable stacking tasks.

C. Batch Renderer Optimization

Rendering thousands of high-fidelity 3DGS scenes simultaneously presents a significant memory challenge. To optimize memory usage while maintaining visual fidelity, we propose several key advancements as follows:

Efficient Pruning Strategy. We adopt an efficient pruning strategy inspired by recent works [15, 12, 14] to reduce the number of Gaussians while preserving visual quality for robot-learning workloads. For static scenes, our pruned representation keeps 30% of the original Gaussians while maintaining high PSNR and SSIM (Table IV); for dynamic objects and robot-linked assets, pruning can be more aggressive depending on visual complexity. This reduction significantly lowers VRAM usage and improves rendering speed, making the renderer well-suited for large-scale, high-fidelity simulations.

Throughput Scaling. We build our Batch-3DGS rendering pipeline on top of gsplat [55] and introduce simulation-oriented system optimizations for large-scale parallel environments. With these optimizations, we can render up to 2048 scenes at a resolution of 640×480 with a total throughput of up to 10,000 FPS; in single-environment rendering of the same scene, our pruned 3DGS reaches 648 FPS versus 125 FPS for vanilla 3DGS. This scaling significantly improves throughput per unit compute, supporting large-batch training workflows.

Rigid-Link Gaussian Kinematics (RLGK). To ensure temporal consistency and eliminate visual artifacts in dynamic scenarios, we introduce Rigid-Link Gaussian Kinematics (RLGK), which binds clusters of 3D Gaussians to corresponding rigid bodies in the physics engine. This coupling ensures that visual representations move synchronously with their physical counterparts, enabling "zero-overhead" updates and artifact-free dynamic rendering during fast motion or contact events.

TABLE II: **Comparison of LiDAR simulation capabilities**.

LiDAR Sensor Feature	Ours	IsaacSim	Gazebo
Rotating LiDAR Support	✓	✓	✓
Solid-State LiDAR Support	✓	✓	✓
Non-repetitive scan LiDAR	✓	×	✓
Static Irregular Objects	✓	✓	✓
Dynamic Irregular Objects	✓	×	×
Self-Occlusion	✓	×	×
3DGS Representation	✓	×	×
Massively Parallel	✓	✓	×

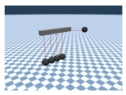
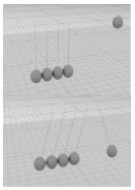
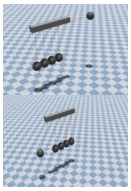
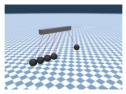
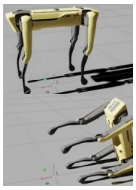
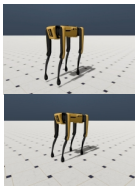
D. Real2Sim Asset Pipeline

Users can create a set of simulation-ready assets from a single RGB image via our Real2Sim pipeline. The pipeline reconstructs 3D representations (3DGS/mesh), estimates geometric pose and scale, and supports task-level physical parameters for collision modeling and manipulation learning. In our experiments, mass, friction, and inertia are assigned through task configuration, manual tuning, or domain randomization rather than estimated from the input image; the modular pipeline can incorporate system identification when interaction data is available.

Objects Segmentation and Background Inpainting. We present an automated pipeline for object segmentation and background inpainting from a single RGB image. Objects are detected using Grounding DINO [32] and segmented with SAM1/SAM2 [26, 41] under a prompt-wise independent detection scheme, enabling explicit tracking of instance–label associations. To mitigate unreliable semantic similarity in open-vocabulary detection, we de-duplicate instances using visual criteria only: mask IoU for general redundancy and a dual-criterion rule based on mask inclusion and boundary overlap to correct over-segmentation. Instance selection is prioritized by a composite confidence score combining detection and segmentation confidence. Occluded object regions are recovered through an iterative mask expansion process coupled with sequential inpainting. Objects are removed one at a time, and after each inpainting step, the scene is re-detected to identify newly exposed regions that are spatially adjacent and label-consistent with existing instances under a bounded area growth constraint. Background inpainting is performed sequentially using LaMa [44] to ensure stable reconstruction.

Assets Generation. For object-level assets, we apply SAM-3D [8] to the original RGB image together with the object mask (M_{obj}) to reconstruct its 3DGS and mesh, and to estimate its pose and scale. For scene-level assets, the inpainted background is processed by AnySplat [22] to generate the background 3DGS, depth map (D_{bg}), as well as the camera intrinsics and extrinsics. To align object- and scene-level assets, we first transform the object such that its rendered depth map (D_{obj}) aligns with the background depth (D_{bg}). The object is then scaled so that the area of its rendered mask matches the original object mask (M_{obj}), measured by pixel occupancy. To reduce memory footprint for downstream robotic tasks, we apply Speedy- splat [14] for 3DGS pruning.

TABLE III: **Qualitative comparison of contact dynamics**. The Newton's Cradle case evaluates momentum transfer, while the Boston Dynamics Spot case validates base stability with a 10 ms timestep.

Scenario	MuJoCo	IsaacSim	Ours
Newton's Cradle (pre-impact)			
Newton's Cradle (post-impact)			
Boston Spot (t=0s)			
Boston Spot (t=2s)			

E. User Interface and Ecosystem Design

`GS-Playground` provides a rich set of features including multi-modal sensing, seamless ecosystem compatibility, and cross-platform support to facilitate robot development.

Multi-modal Sensor Suite Beyond standard RGB and depth streams, we integrate a high-performance Batch-LiDAR module utilizing ray-casting to generate high-fidelity point clouds and heightmap scanning (Table II). Additionally, our platform provides detailed contact information equivalent to MuJoCo, including multi-point contact forces, torques, and decomposed normal/tangential components.

Ecosystem Compatibility and Cross-Platform Workflow. Prioritizing "zero-friction" migration, our API is compatible with the MuJoCo MJCF format, enabling rapid project migration. The physics engine features a cross-platform architecture (Windows/Linux/macOS), allowing local prototyping and debugging before deploying large-scale parallel training on Linux GPU clusters using Batch-3DGS and CUDA-optimized perception modules.

IV. Results

We evaluate `GS-Playground` through a comprehensive benchmark suite spanning visual and geometric fidelity, physics stability, manipulation proficiency, and locomotion capability. Our results demonstrate the platform's superiority in photorealistic rendering, massive parallel physics stepping, and effective Sim2Real transfer for both vision-based and contact-rich tasks.

A. Physics Stability and Solver Robustness

Multi-Scenario Benchmarking. We conduct a multi-scenario stability study across multiple simulators to stress-test contact handling under challenging dynamics.

1) *Hard Contact & Momentum Conservation*: Using a Newton's Cradle setup with identical initial perturbations across engines, we evaluate long-horizon momentum transfer and dissipation under hard contacts (Table III Top). `GS-Playground` preserves impact timing and swing amplitude with reduced energy bleed across repeated impacts, while MuJoCo exhibits stronger damping and phase drift.

2) *Large Time-step Stability*: We evaluate base stability on a Boston Dynamics Spot model under physics-only stepping with a 10 ms timestep, no control input, and identical initial pose (Table III Bottom). `GS-Playground` exhibits smaller base displacement and reduced drift over time, suggesting more stable contact resolution under large time steps.

3) *Complex Multi-Body Interactions*: In a dense store shelf scenario with stacked objects, we evaluate static stability under complex multi-contact constraints. While `GS-Playground` consistently converges to stable equilibria, MuJoCo exhibits characteristic jitter and contact-induced drift—common artifacts in high-density contact graphs (Fig. 3).

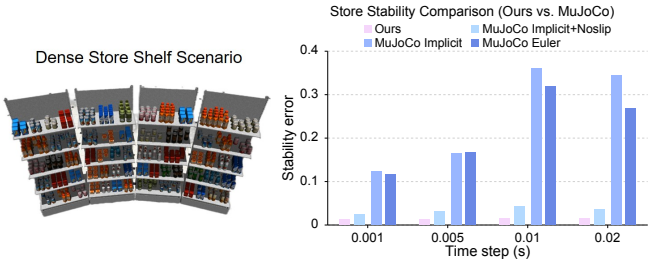


Fig. 3: **Physics stability under complex multi-body interactions**. (a) The dense store shelf scenario; (b) Stability error across time steps, computed over all objects in the scene with identical initial placements. The error is defined as $\sqrt{\Delta p^2 + \Delta\theta^2}$, where Δp is mean positional drift (m) and $\Delta\theta$ is mean orientation drift (rad).

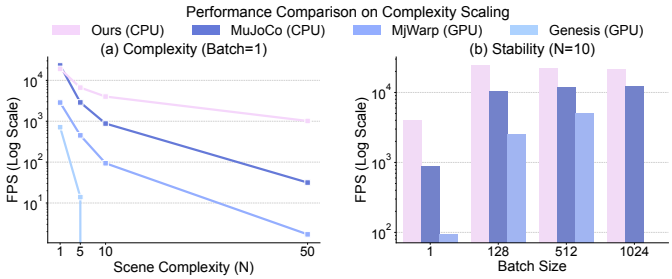


Fig. 4: **Performance Comparison on Complexity Scaling.** Left: As the complexity (N, the number of humanoid robots) in a single environment increases, the FPS advantage of our framework becomes increasingly pronounced. Right: At a complexity of $N = 10$, our framework maintains FPS advantage with large batch sizes, where Genesis fails to reach convergence.

4) *Stability and Scalability in Complex Scenes.* We evaluate the algorithmic robustness of `GS-Playground` against MuJoCo (CPU) and Genesis/MjWarp (GPU) in high-complexity single-environment scenarios. To vary scene complexity, we scale the number (N) of 27-DoF humanoid agents within a single environment. All experiments were conducted on an AMD 9950x CPU and an NVIDIA RTX 5090 GPU. Results are shown in Fig. 4. As constraint density increases, GPU-based solvers suffers from severe performance degradation. At $N = 10$, Genesis fails to reach convergence and exhibits numerical instability through Jacobian-related errors. When complexity reaches $N = 50$, MjWarp's throughput collapses to a mere 1.71 FPS. In contrast, `GS-Playground` (CPU) maintains a robust throughput of 1,015 FPS at $N = 50$. This perfor-

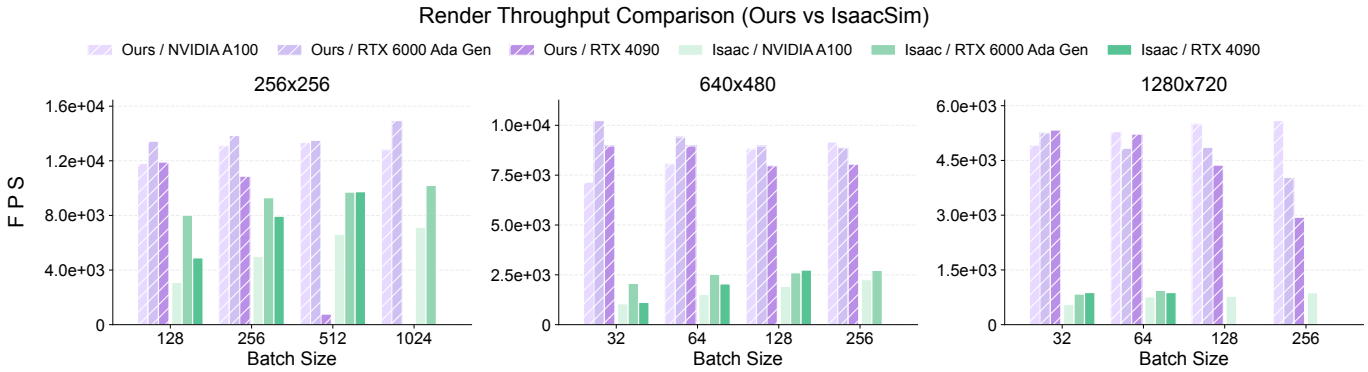


Fig. 5: **Aggregate rendering throughput of GS-Playground vs. Isaac Sim's ray-tracing renderer across resolutions.** Isaac Sim relies on manual asset modeling and encounters Out-Of-Memory (OOM) exceptions at higher resolutions, while GS-Playground leverages automated asset generation from real-world captures. Evaluations span three GPU architectures: NVIDIA RTX 4090, RTX 6000 Ada Generation, and NVIDIA A100.

TABLE IV: **Image Quality Metrics.** Our pruned model retains 30% of the original Gaussians while preserving high PSNR and SSIM for static scene reconstruction.

Method	# Gaussians ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	100%	27.1503	0.8296	0.2238
Ours	30%	26.8658	0.8022	0.2840

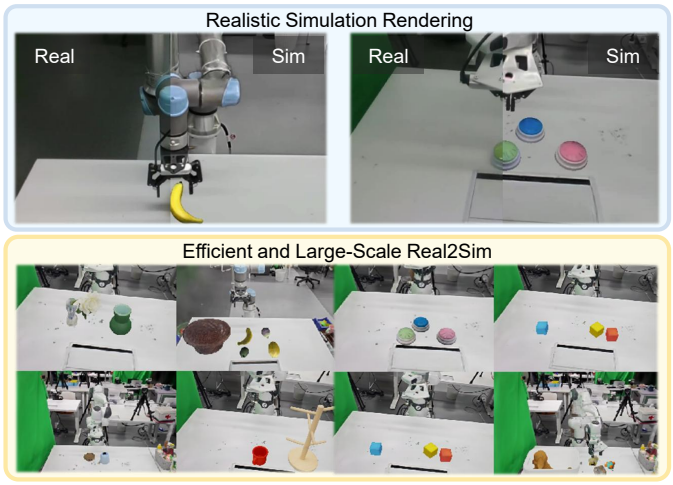


Fig. 6: **Visualization of rendering**. The simulated renderings are nearly indistinguishable from real photographs indicating photorealistic fidelity. Our framework supports a broad range of tasks and evaluations, including diverse objects and scene configurations.

mance represents a 32× speedup over MuJoCo and a ∼600× improvement over MjWarp. Note that `GS-Playground` also offers competitive GPU performance, with further gains expected as we are refining our GPU-specific kernel fusion and memory management strategies.

B. Visual Fidelity and Efficiency

Visual 3DGS Compression. We measure the trade-off between memory footprint and visual quality, as summarized in Table IV. For full static scene reconstruction, we compare raw 3DGS reconstructions with our pruned version, retaining only about 30% of the Gaussians while preserving high PSNR and SSIM. Additionally, for manipulated dynamic objects and the robot body, the number of points can be further reduced by up to 90%, enabling more aggressive memory compression without compromising the critical visual cues required for robotic perception.

Rendering Throughput Comparison. We evaluate the rendering throughput of `GS-Playground` against Isaac Sim's ray-tracing renderer across three standard resolutions and multiple GPU architectures, including the NVIDIA RTX 4090, RTX 6000 Ada, and A100. As illustrated in Fig. 5, our framework consistently outperforms the baseline in aggregate FPS at all tested batch sizes, with the gap most pronounced at higher resolutions such as 1280×720, where Isaac Sim's ray-tracing approach frequently encounters Out-Of-Memory (OOM) exceptions at larger batch sizes. Our Gaussian Splatting-based pipeline thus provides a more memory-efficient and scalable solution for large-scale parallel visual simulation.

Qualitative Rendering Fidelity and Diversity. Fig. 6 presents a qualitative comparison between real and simulated renderings. The simulated renders exhibit a high degree of visual consistency with real camera images, preserving critical geometric features and surface details. Moreover, the scenes depicted span a diverse set of object types, materials, and configurations, demonstrating that our simulation system maintains a high level of visual consistency across varied setups, making it suitable for diverse training and evaluation scenarios. All experiments were conducted on the Bridge-v2 dataset [47]. On an NVIDIA RTX 3090 GPU, the per-scene pipeline latency is within 5 minutes end-to-end before 3DGS pruning, while dataset-scale generation distributes pipeline stages across multiple GPUs. Compared to Bridge-v2, BridgeGS enriches each asset with scene- and object-level 3DGS representations, object-level meshes, object poses, and camera intrinsics and extrinsics; Sec. B.1 provides additional dataset details.

C. Physics-Intensive Locomotion Learning

Simulation Comparison. We benchmark our framework against IsaacLab using the *Isaac-Velocity-Flat-Unitree-Go1-v0* and *Isaac-Velocity-Rough-Unitree-Go1-v0* environments. All configurations are trained for an equivalent number of total

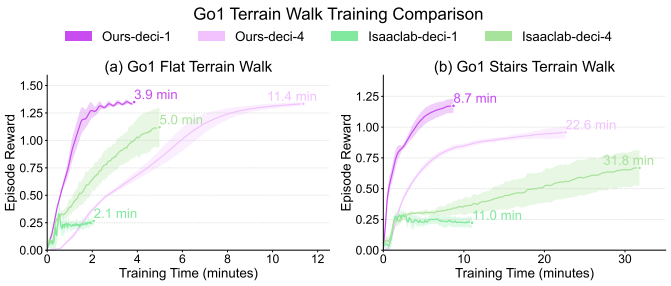


Fig. 7: **Wall-clock training efficiency for Unitree Go1 locomotion.** (a) Flat terrain; (b) Rough terrain (stairs). "deci" denotes the decimation, which refers to the number of physical sub-steps per control step. Lower decimation typically increases throughput but may compromise physical fidelity. All results—both `GS-Playground` and the IsaacLab baseline—were measured on identical hardware (RTX 5070Ti + Ryzen 7 9700X).

steps. On flat terrain (Fig. 7a), while IsaacLab achieves higher speed at low fidelity ($d = 1$, where d denotes the decimation factor, or the number of physical sub-steps per control step), it fails to reach a competitive terminal reward. In contrast, our method with $d = 1$ achieves a terminal reward comparable to IsaacLab at $d = 4$, while reaching convergence faster. This result demonstrates that our solver's stability permits larger integration time steps without compromising physical fidelity or policy convergence. This stability advantage is even more apparent in complex environments like the stairs terrain (Fig. 7b). Here, our framework at $d = 1$ achieves higher rewards and faster convergence than the baseline at the same setting. Against the high-precision baseline ($d = 4$), our method remains faster in wall-clock time. These results show that our approach offers a clear dual advantage in stability and speed for complex, contact-rich tasks.

Sim2Real Deployment. We demonstrate the practical utility of `GS-Playground` by successfully deploying state-based locomotion policies onto a Unitree Go2 quadruped and a Unitree G1 humanoid (Fig. 8(a), (b)). The quadruped policy, utilizing simplified collision geometries and 1,024 parallel environments, reached convergence in 10 minutes of wall-clock time. The humanoid policy, employing full-collision manifolds and 2,048 parallel environments, reached convergence in approximately 6 hours. These results demonstrate our framework's efficiency in bridging the physics reality gap.

D. Vision-Centric Navigation Learning

We demonstrate a vision-based navigation task on the Unitree Go2, where the robot is required to search for and reach a target traffic cone (Fig. 8(d)). The policy observes egocentric RGB images rendered by `GS-Playground` during training. To couple high-level visual decision making with stable legged control, we adopt a two-level hierarchical RL design. A *high-level* policy encodes egocentric RGB observations and outputs a compact navigation command (e.g., desired base motion command). A *low-level* locomotion controller takes the command together with proprioceptive states and produces joint-level control signals for the Go2. Both levels use PPO: the low-level controller is pre-trained and frozen, while the high-level policy is trained in `GS-Playground`. After training in simulation, we directly deploy the learned policy on a real Go2, which successfully performs goal-directed navigation toward the cone using only onboard egocentric vision. These results validate that `GS-Playground` provides the high-fidelity visual feedback necessary for training vision-encoder policies capable of zero-shot real-world deployment.

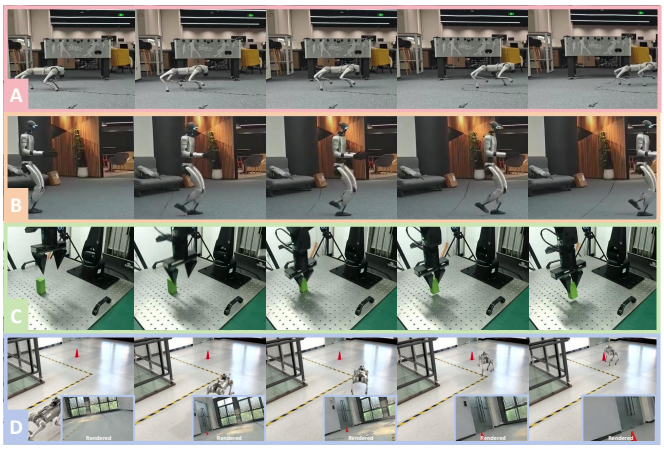


Fig. 8: **Real-world deployment of policies trained in GS-Playground**. We demonstrate robust Sim2Real transfer across diverse embodiments and modalities: **(a) Quadrupedal Locomotion**: Velocity tracking on Unitree Go2; **(b) Humanoid Locomotion**: 23-DoF balancing and walking on Unitree G1; **(c) Visual Manipulation**: End-to-end RGB-based grasping; **(d) Visual Navigation**: Real-time cone following on Unitree Go2 using raw RGB observations.

E. Vision-Centric Manipulation Learning

To evaluate the platform's efficacy in bridging the visual Sim2Real gap, we conducted a block-grasping task using the Airbot Play robotic arm (Fig. 8c). The control policy, trained as a goal-conditioned RL agent, maps raw RGB observations and proprioceptive states to 6-DoF joint actions. Utilizing our Real2Sim pipeline, we reconstructed a high-fidelity 3DGS simulation scene that serves as a direct visual proxy for the real-world setup. To improve robustness against real-world variability, we incorporated domain randomization of camera poses and image-level appearance (brightness, contrast, and exposure) during training. For comparison, we trained the same policy under matched task definition, network, PPO hyperparameters, and randomization protocol in three batched-rendering baselines—Mujoco Playground, ManiSkill3, and Isaac Lab—while varying the simulator and visual rendering stack. The resulting policy achieves an 18/20 success rate (90%) during zero-shot real-world deployment, whereas all three baselines achieve 0/20 under the same real-world evaluation protocol (Table V and Fig. 9). Under this matched setup, the observed failures are consistent with a visual domain gap between the baseline renderings and the real scene, rather than a difference in task definition or policy architecture. Notably, the evaluation was performed in a real-world scene that featured no specialized visual engineering or simplification, such as simplified backgrounds or controlled lighting. The robot demonstrates agile and stable grasping maneuvers, indicating

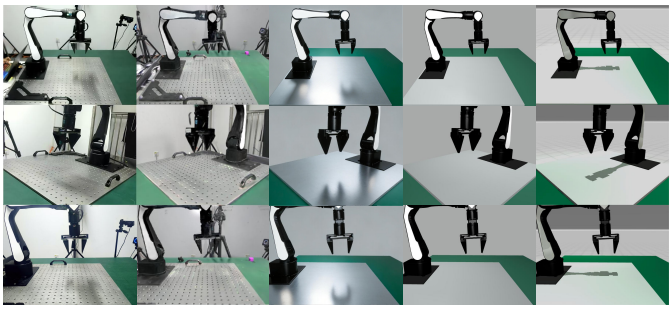


Fig. 9: **Rendering comparison for the AIRBOT PickCube scene.** From left to right: Real World, `GS-Playground`, Isaac Lab, ManiSkill3, and Mujoco Playground. `GS-Playground` closely matches the appearance of the real scene, while all other simulators exhibit a substantial visual domain gap that, under the matched setup of Table V, leads to 0% Sim2Real transfer.

TABLE V: **Zero-shot Sim2Real success rate on PickCube** under matched task setup, network, and image-level domain randomization (20 real-world trials each).

Simulator	Real-world Success Rate
Mujoco Playground	0 % (0/20)
ManiSkill3	0 % (0/20)
Isaac Lab	0 % (0/20)
`GS-Playground` (Ours)	**90 % (18/20)**

that the perceptual richness of 3D-Gaussian-based simulation can support policies trained from complex, unsimplified visual cues. Sec. D provides additional implementation details.

V. Discussion

We plan to utilize `GS-Playground` to synthesize large-scale vision-informed data for VLA and VLN models, facilitating robust sim-to-real transfer. Additionally, by incorporating the real-to-sim workflows, we are constructing expansive, scalable environments for the rigorous verification and benchmarking of advanced robotic policies.

Despite its current performance, our framework has several limitations that provide opportunities for future research. `GS-Playground` currently focuses on interactive rigid-body robot learning across manipulation, locomotion, and navigation. More general Real2Sim, including automatic estimation of physical parameters from visual observations alone, remains future work.

Unlike ray tracing or standard rasterization-based renderers, 3D Gaussian Splatting bakes static lighting into the reconstructed scene. Visual augmentation such as color jitter and brightness randomization improves robustness, but does not fully solve dynamic lighting. Transparent or highly reflective objects can also remain difficult even with segmentation and inpainting, especially under severe transparency, specular reflections, or heavy occlusion.

Furthermore, the current Rigid-Link Gaussian Kinematics (RLGK) assumes rigid bodies. Representing deformable objects (cloth, fluids) or soft-body manipulation remains a challenge. We plan to integrate particle-based dynamics (like PBD or MPIM) with Gaussian splatting to address non-rigid interactions.

VI. Conclusion

We introduce `GS-Playground`, a high-performance simulation platform that harmonizes a custom parallel physics engine with a memory-efficient batch 3D Gaussian Splatting (3DGS) renderer to bridge the gap between realism and efficiency. By utilizing a specialized point-pruning strategy, our framework achieves an aggregate rendering throughput of 10^4 FPS across 2048 environments at 640×480 resolution while avoiding the heavy memory overhead of traditional neural rendering. An automated Real2Sim pipeline streamlines the creation of simulation-ready visual and geometric assets for complex robot-learning scenes. Evaluations across quadrupedal, humanoid, and manipulation embodiments demonstrate that `GS-Playground` facilitates robust Sim2Real transfer by bridging the reality gap in both physical dynamics and visual perception. Together, these capabilities establish `GS-Playground` as a scalable, open-source platform for high-throughput photorealistic robot learning and Sim2Real research.

VII. Acknowledgments

The authors gratefully acknowledge support from D-Robotics under Grant No. 20243000104. We thank the MuJoCo and MuJoCo Playground teams for making their codebases available to the community; their software and documentation provided valuable references and infrastructure support for this work.

References

[1] Jad Abou-Chakra, Lingfeng Sun, Krishan Rana, Brandon May, Karl Schmeckpeper, Maria Vittoria Minniti, and Laura Herlant. Real-is-sim: Bridging the sim-to-real gap with a dynamic digital twin for real-world robot policy evaluation. *arXiv preprint arXiv:2504.03597*, 2025.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[3] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. *arXiv preprint arXiv:2412.14957*, 2024.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. NavDP: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2501.04610*, 2025.

[6] Zhanxiang Cao, Yang Zhang, Buqing Nie, Huangxuan Lin, Haoyang Li, and Yue Gao. Learning motion skills

with adaptive assistive curriculum force in humanoid robots. *arXiv preprint arXiv:2506.23125*, 2025.

[7] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.

[8] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[9] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *RSS*, 2025.

[10] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[11] Alejandro Escontrela, Justin Kerr, Arthur Allshire, Jonas Frey, Rocky Duan, Carmelo Sferrazza, and Pieter Abbeel. Gaussgym: An open-source real-to-sim framework for learning locomotion from pixels. *arXiv preprint arXiv:2510.15352*, 2025.

[12] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024.

[13] Xiaoshen Han, Minghuan Liu, Yilun Chen, Junqiu Yu, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^3 sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation. *arXiv preprint arXiv:2502.08645*, 2025.

[14] Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21537–21546, 2025.

[15] Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5949–5958, 2025.

[16] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[17] Tairan He, Zi Wang, Haoru Xue, Qingwei Ben, Zhengyi Luo, Wenli Xiao, Ye Yuan, Xingye Da, Fernando Castañeda, Shankar Sastry, et al. Viral: Visual sim-to-real at scale for humanoid loco-manipulation. *arXiv preprint arXiv:2511.15200*, 2025.

[18] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[19] Yufei Jia, Guangyu Wang, Yuhang Dong, Junzhe Wu, Yupei Zeng, Haonan Lin, Zifan Wang, Haizhou Ge, Weibin Gu, Kairui Ding, et al. Discoverse: Efficient robot simulation in complex high-fidelity environments. *arXiv preprint arXiv:2507.21981*, 2025.

[20] Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025.

[21] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *arXiv preprint arXiv:2503.17973*, 2025.

[22] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

[23] Yongpeng Jiang, Mingrui Yu, Chen Chen, Yongyi Jia, and Xiang Li. Robust in-hand reorientation with hierarchical rl-based motion primitives and model-based regrasping. *IEEE Robotics and Automation Practice*, 1: 12–17, 2025.

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

[25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[26] 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. *arXiv:2304.02643*, 2023.

[27] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[28] Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.

[29] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024.

[30] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu,

Wei Liang, Yixin Zhu, and Siyuan Huang. CLONE: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. In *9th Annual Conference on Robot Learning (CoRL)*, 2025.

[31] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. *arXiv preprint arXiv:2501.16764*, 2025.

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[33] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, et al. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15379–15386. IEEE, 2025.

[34] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[35] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[36] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[37] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning (CoRL)*, pages 734–743. PMLR, 2018.

[38] Lars Mescheder, Wei Dong, Shiwei Li, Xuyang Bai, Marcel Santos, Peiyun Hu, Bruno Lecouat, Mingmin Zhen, AmaÃĢl Delaunoy, Tian Fang, et al. Sharp monocular view synthesis in less than a second. *arXiv preprint arXiv:2512.10685*, 2025.

[39] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[40] M Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6502–6509. IEEE, 2025.

[41] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[42] Brennan Shacklett, Luc Guy Rosenzweig, Zhiqiang Xie, Bidipta Sarkar, Andrew Szot, Erik Wijmans, Vladlen Koltun, Dhruv Batra, and Kayvon Fatahalian. An extensible, data-oriented architecture for high-performance, many-world simulation. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023.

[43] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[44] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.

[45] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025.

[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[47] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[48] 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*, pages 5294–5306, 2025.

[49] Zifan Wang, Yufei Jia, Lu Shi, Haoyu Wang, Haizhou Zhao, Xueyang Li, Jinni Zhou, Jun Ma, and Guyue Zhou. Arm-constrained curriculum learning for loco-manipulation of a wheel-legged robot. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10770–10776. IEEE, 2024.

[50] Zifan Wang, Teli Ma, Yufei Jia, Xun Yang, Jiaming Zhou, Wenlong Ouyang, Qiang Zhang, and Junwei Liang. Omni-perception: Omnidirectional collision avoidance

for legged locomotion in dynamic environments. *arXiv preprint arXiv:2505.19214*, 2025.

[51] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time interactive realistic and browser-compatible environment from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4588, 2024.

[52] Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. Vid2sim: Realistic and interactive simulation from video for urban navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1581–1591, 2025.

[53] Haoru Xue, Tairan He, Zi Wang, Qingwei Ben, Wenli Xiao, Zhengyi Luo, Xingye Da, Fernando Castañeda, Guanya Shi, Shankar Sastry, et al. Opening the sim-to-real door for humanoid pixel-to-action policy transfer. *arXiv preprint arXiv:2512.01061*, 2025.

[54] Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025.

[55] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

[56] Zhao-Heng Yin, Changhao Wang, Luis Pineda, Francois Hogan, Krishna Bodduluri, Akash Sharma, Patrick Lancaster, Ishita Prasad, Mrinal Kalakrishnan, Jitendra Malik, et al. DexterityGen: Foundation controller for unprecedented dexterity. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[57] Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2render2real: Scaling robot data without dynamics simulation or robot hardware. *arXiv preprint arXiv:2505.09601*, 2025.

[58] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025.

[59] Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*, 2025.

[60] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024.

[61] Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.

[62] Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. *arXiv preprint arXiv:2511.04665*, 2025.

[63] Yang Zhang, Zhanxiang Cao, Buqing Nie, Haoyang Li, Zhong Jiangwei, Qiao Sun, Xiaoyi Hu, Xiaokang Yang, and Yue Gao. Keep on going: Learning robust humanoid motion skills via selective adversarial training. *arXiv preprint arXiv:2507.08303*, 2025.

[64] Xian Zhou, Yiling Qiao, Zhenjia Xu, TH Wang, Z Chen, J Zheng, Z Xiong, Y Wang, M Zhang, P Ma, et al. Genesis: A generative and universal physics engine for robotics and beyond. *arXiv preprint arXiv:2401.01454*, 2024.

[65] Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. *arXiv preprint arXiv:2505.21906*, 2025.

[66] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[67] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.