DISCOVERSE: Efficient Robot Simulation in Complex High-Fidelity Environments

Yufei Jia[1,†], Guangyu Wang[1,†], Yuhang Dong[2], Junzhe Wu[1], Yupei Zeng[2], Haonan Lin[3], Zifan Wang[4], Haizhou Ge[1], Weibin Gu[1], Kairui Ding[1], Zike Yan[1], Yunjie Cheng[5], Yue Li[7], Ziming Wang[6], Chuxuan Li[1], Wei Sui[8], Lu Shi[1], Guanzhong Tian[2], Ruqi Huang[1,‡], Guyue Zhou[1,‡]

***Abstract*— We present DISCOVERSE, the first unified, modular, open-source 3DGS-based simulation framework for Real2Sim2Real robot learning. It features a holistic Real2Sim pipeline that synthesizes hyper-realistic geometry and appearance of complex real-world scenarios, paving the way for analyzing and bridging the Sim2Real gap. Powered by Gaussian Splatting and MuJoCo, DISCOVERSE enables massively parallel simulation of multiple sensor modalities and accurate physics, with inclusive supports for existing 3D assets, robot models, and ROS plugins, empowering large-scale robot learning and complex robotic benchmarks. Through extensive experiments on imitation learning, DISCOVERSE demonstrates state-of-the-art zero-shot Sim2Real transfer performance compared to existing simulators. For code and demos: https://air-discoverse.github.io/.**

I. INTRODUCTION

End-to-end learning has emerged as a scalable, efficient, and cost-effective solution for robotics, enabling direct policy learning from raw sensor data. This paradigm underscores the crucial need for fast and robust simulators, a research area that has witnessed rapid advancements in recent years [1], [2], [3], [4], [5], [6], [7], [8], [9]. However, a critical challenge in end-to-end learning remains largely unsolved, i.e., the dramatic performance degradation when transferring from simulation to the real world, a.k.a. the Sim2Real gap [10], which fundamentally originates from the visual discrepancies between simulation and reality [11]. In simulation, the appearance is typically rendered from artificial assets with handcrafted textures and simplified lighting, failing to faithfully characterize the complexity of the real world.

To circumvent this issue, some Real2Sim approaches [12], [13], [14], [15] leverage 3D reconstruction techniques to build virtual replicas of the real world, with the aim to inherit the rich appearance, structures, and semantics from reality. However, they are mainly designed for navigation-oriented tasks and primarily lack photorealism. The reason is that traditional multi-view stereo (MVS) and RGB-D fusion approaches are vulnerable to non-Lambertian reflectance and thin geometry, inevitably leading to large amounts of collapsed surface and thus severely deteriorated visual quality. On the other hand, state-of-the-art simulator Omniverse Issac Lab [3] enables high-quality Physically-Based Rendering (PBR) in real-time through GPU acceleration. Despite the notable progress, they need onerous configurations and lack supports for Real2Sim assets, especially for those with large amounts of primitives. Very recently, there are several early attempts [16], [17], [18] that utilize advanced neural representations – e.g., 3D gaussian splatting (3DGS) [19] – to build Real2Sim replicas as radiance fields. However, they fail to recover precise geometry and relightable appearance, exhibit poor view extrapolation capabilities, and lack robustness in complex real-world scenarios. For example, they struggle to handle in-the-wild scenes with intricate geometry, textures, and illumination, large-scale scenes with unstructured and sparse imagery, and also textureless or non-Lambertian surfaces. Therefore, they are unsuitable as general-purpose robotic simulators for diverse real-world applications.

In light of these observations, we introduce DISCOVERSE, the first unified, modular, open-source 3DGS-based simulation framework with a collection of novel features to facilitate end-to-end robotic solutions:

1) High-fidelity, hierarchical Real2Sim generation for both background node and interactive scene nodes in various complex real-world scenarios, leveraging advanced laser-scanning, generative models, physically-based relighting, and Mesh-Gaussian transfer.
2) Efficient simulation and user-friendly configuration. By seamlessly integrating 3DGS rendering engine, MuJoCo physical engine, and ROS2 robotic interface, we provide an easy-to-use, massively parallel implementation for rapid deployment and flexible extension. We also propose an automated state generation approach to facilitate demonstration collection. The overall throughput achieves 650 FPS for 5 cameras rendering RGB-D frames, which is $\sim 3\times$ faster than Issac Lab (ORBIT) [3].
3) Compatibilities with existing 3D assets and inclusive supports for robot models (robotic arm, mobile manipulator, quadrocopter, etc.), sensor modalities (RGB, depth, LiDAR, tactile sensors), ROS plugins, and various randomizations (e.g., generative-based).

[1]Tsinghua University, [2]Zhejiang University, [3]Huazhong University of Science and Technology, [4]Hong Kong University of Science and Technology (Guangzhou), [5]Xi'an Jiaotong University, [6]Tongji University, [7]DISCOVER Robotics, [8]D-Robotics.

[†]Yufei Jia and Guangyu Wang contributed equally to this work (email: {jyf23, wanggy24}@mails.tsinghua.edu.cn).

[‡]Corresponding authors: Ruqi Huang; Guyue Zhou (email: ruqihuang@sz.tsinghua.edu.cn; zhouguyue@air.tsinghua.edu.cn).

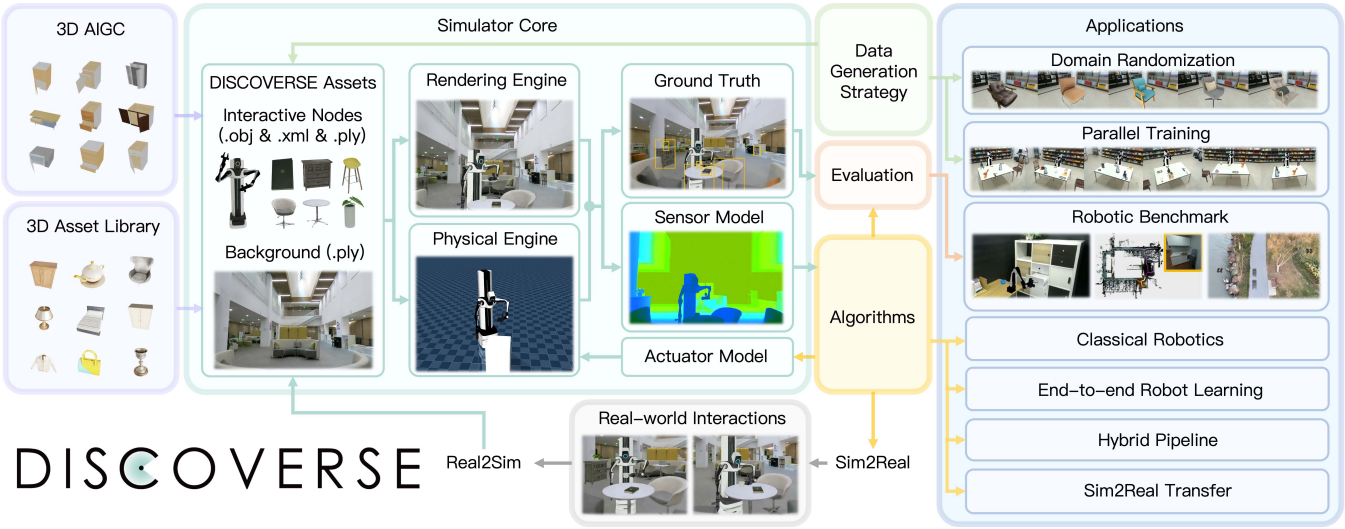

Fig. 1. DISCOVERSE system overview. DISCOVERSE unifies real-world captures, 3D AIGC, and any existing 3D assets in formats of 3DGS (.ply), mesh (.obj/.stl), and MJCF physical models (.xml), enabling their use as interactive scene nodes (objects and robots) or the background node. We leverage Gaussian splatting as our rendering engine to generate hyper-realistic radiance field rendering of multiple sensor modalities and use MuJoCo as the physical engine to ensure accurate physics. Benefiting from the efficiency and fidelity, DISCOVERSE enables user-definable data generation strategy, evaluation metrics, and algorithms for robotics and embodied AI, empowering a variety of applications, e.g., parallel training, complex robotic benchmarks, etc.

A high-level comparison with existing simulators is listed in Table I. Unlike prior works, DISCOVERSE offers a more comprehensive Real2Sim solution that generates high-quality replicas of diverse real-world scenarios with precise geometry and harmonious appearance, thus yielding a more robust and versatile framework for bridging the Sim2Real gap.

To verify the effectiveness of DISCOVERSE, we conduct extensive experiments on imitation learning (IL) using both ACT [20] and Diffusion Policy (DP) [21] across three real-world manipulation tasks. We compare DISCOVERSE against three state-of-the-art simulators – MuJoCo [1], RoboTwin [6], and SplatSim [18] – by fairly deploying ACT and DP on each of them. Our results demonstrate that DISCOVERSE significantly outperforms existing simulators in zero-shot Sim2Real transfer. In particular, we identify the following findings:

1) Using ACT, DISCOVERSE increases the average success rate relative to the second best simulator (SplatSim) by $\sim$11% without data augmentation, and by 18.5% after data augmentation.
2) Similar results are demonstrated with DP, where DISCOVERSE outperforms SplatSim by $\sim$11% without data augmentation, and by 11.4% after augmentation.
3) Image-based augmentation further mitigates domain shifts and improves the average success rate of DISCOVERSE by 31.5% with ACT and by 29.3% with DP.
4) DISCOVERSE enables $\sim$100$\times$ more efficient data collection compared to real-world demonstrations.

These results underscore the great potential of DISCOVERSE in steering Sim2Real towards Real2Real. We believe DISCOVERSE lays solid foundation for comprehensive Sim2Real robotic benchmarks, including manipulation, navigation, multi-agent collaboration, etc., to stimulate further research and practical applications.

II. Related Works

A. Simulation Environments for Robotics

Simulators [1], [2], [3], [4], [5], [6], [7], [8], [9] are crucial in scaling up robot learning by offering efficiency and safety. The underlying physical engines [1], [22] enable high-throughput simulation of accurate contact, collision, and deformation dynamics, empowering a variety of complex robotic tasks [11], [20], [23]. As another core component, the renderer, which synthesizes visual inputs for robots, is commonly based on game engines that are widely deployed in prior frameworks [3], [4], [5], [6], [7], [8]. While game engines are directly compatible with artificial game assets and allow for manual customization of the environments, the process of asset creation can be time-consuming, and the renderings fail to fully characterize real-world richness, thus limiting the scalability, diversity, and fidelity of simulation.

B. Real2Sim2Real Robot Learning

To alleviate domain shifts, some methods [12], [14], [9], [15] propose to leverage scanned 3D mesh of real-world scenes in simulation. RoboTwin [6] uses 3D generation [24] to enable superior object-level Real2Sim quality. Some recent works [16], [17], [18] further adopt 3DGS [19] representation to enable comprehensive 3D reconstruction. Despite these advancements, existing 3DGS-based solutions still face challenges in generalizing to complex real-world scenarios, since high-quality results can only be obtained with ultra-dense multi-view captures of targets featuring rich textures and diffuse reflectance. Without these conditions, the reconstruction tends to suffer from artifacts such as excessive floaters or blurriness. Besides, these simulators reconstruct interactive targets on scene and lack supports for existing mesh assets and appearance relighting, making them inefficient and less practical for general-purpose simulation.

TABLE I

COMPARISON BETWEEN DISCOVERSE AND OTHER SIMULATORS SUPPORTING END-TO-END ROBOT LEARNING.

Simulators	3D Representation	Physics Engine	Renderer	Scene-level Real2Sim	Scene-level Fidelity	†Scene Complexity	Object-level Real2Sim	Object-level Fidelity
Matterport3D [12]	Mesh	None	MeshRender	✓	⋆	H	×	⋆
SAPIEN [4]	Mesh	PhysX4	OpenGL	×	⋆	H	×	⋆
ThreeDWorld [5]	Mesh	PhysX4/FleX	Unity3D	×	⋆	THW	×	⋆
ManipulatorThor [8]	Mesh	PhysX4	Unity	×	⋆	H	×	⋆
iGibson [15]	Mesh	Bullet	OpenGL	✓	⋆	H	×	⋆
Habitat2.0 [2]	Mesh	Bullet	Magnum	×	⋆	H	×	⋆
ManiSkill2 [7]	Mesh	PhysX4/Warp	OpenGL	×	⋆	TH	×	⋆
ORBIT [3] (Issac Lab)	Mesh	PhysX5.1	Omni.RTX	×	⋆	THW	×	⋆
RoboTwin [6]	Mesh	PhysX4	OpenGL	×	⋆	T	✓	⋆⋆
SplatSim [18]	3DGS	Bullet	Splatting	✓	⋆⋆	T	✓	⋆⋆
DISCOVERSE (Ours)	3DGS	MuJoCo	Splatting	✓	⋆⋆⋆	THW	✓	⋆⋆⋆

† For Scene Complexity, 'T' stands for table-top, 'H' stands for house-scale, 'W' stands for in-the-wild large-scale scenes.

III. SYSTEM ARCHITECTURE

Our goal is to integrate an advanced neural renderer, a state-of-the-art physical engine, and a user-friendly robotic interface into a unified, modular framework, to support various end-to-end robotic perception and interaction tasks (Fig. 1). The simulator is developed with highly optimized implementations and offers Python API to enable rapid deployments and flexible user-driven extensions.

In the following, we introduce the key components of our simulation framework, focusing on two main perspectives: the DISCOVERSE engine and DISCOVERSE asset. We then report our simulation throughput accordingly.

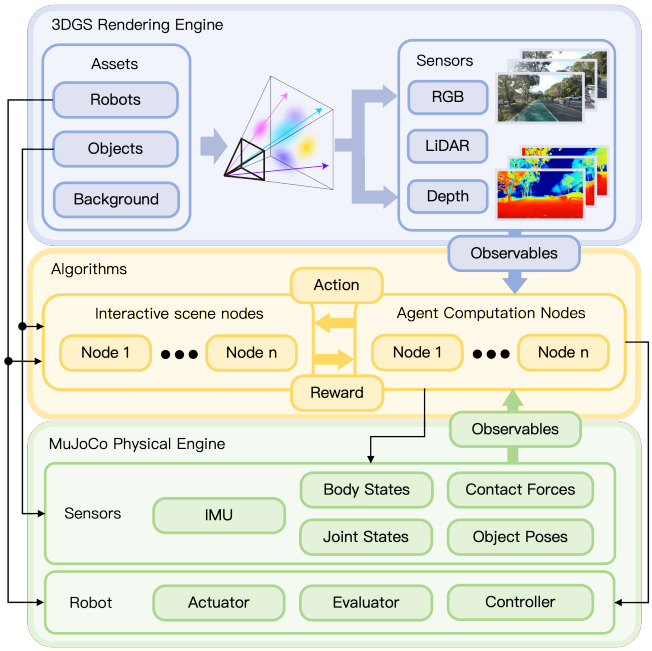

Fig. 2. DISCOVERSE operation flow. We utilize fast tile-based splatting for high-fidelity neural rendering and integrate MuJoCo [1] physical simulator for various robotic utilities.

A. Engine

We adopt the tile-based renderer in 3D Gaussian Splatting [19] to simulate high-fidelity visuals and use the open-source MuJoCo [1] physical engine for accurate robot-object interactions. Our system also includes easy-to-use ROS2 (Robot Operating System 2) [25] support, to facilitate seamless integration and enhance robotic research workflows. The overall operation flow of DISCOVERSE is depicted in Fig. 2.

Renderer. To strike a good balance between efficiency and fidelity, we use the native 3DGS renderer [19] in DISCOVERSE, which features a fast tile-based Gaussian rasterizer with efficient sorting, splatting, spherical harmonics evaluation, and alpha-blending. All of the rendering operations are implemented as highly-optimized custom CUDA kernels to maximize GPU parallelization.

Physical Simulator. We choose MuJoCo [1] for rigid body simulation due to its efficiency and accuracy in handling complex physical interactions, including contacts, friction, and soft constraints. Leveraging MuJoCo's capabilities, DISCOVERSE supports precise force control, P-D control, and inverse dynamics, enabling accurate collision detection, contact force simulation, and the modeling of articulated robot arms or grippers with various joint types under both kinematic and dynamic control.

ROS2 Interface. ROS2 (Robot Operating System 2) [25] is a modular and flexible framework for building robotic applications, offering advanced real-time capabilities, scalability, security, and interoperability. The ROS2 interface in DISCOVERSE offers a set of APIs for interacting with robots using physics. Specifically, we apply encoder torques on joints for low-level control and provide joint space and Cartesian coordinate space APIs for higher-level control, thus bridging the gap between simulation and real-world robotic applications.

B. Asset

High-quality digital assets are central to simulation. To seamlessly integrate the Gaussian splatting renderer with the mesh-based MuJoCo physical engine while ensuring compatibility with existing asset libraries, we introduce a comprehensive Real2Sim pipeline (detailed in Sec. IV). With this functionality, we can easily incorporate real-world captures, 3D AIGC, and existing 3D asset libraries into DISCOVERSE. Additionally, we provide full support for a variety of robot models tailored to different downstream tasks.

Real-world Captures. Our simulator supports multi-view captures of real-world scenes and objects, which are processed using the established workflows of COLMAP [26] and 3DGS [19], [18]. We also have extended supports for laser scanners, so as to enable more robust and accurate reconstructions via the proposed Real2Sim functionality.

AIGC and Asset Library. We ensure compatibilities with recent state-of-the-art 3D AIGC techniques, including textured mesh generation [27], [24], native 3DGS generation [28], and the generation of articulated objects. We also have supports for public 3D datasets, e.g., ShapeNet [29], PartNet [30], Objaverse [31], etc. The universal support for existing 3D assets makes DISCOVERSE a versatile robotic simulator for diverse application scenarios.

Robot Models. We offer a wide range of fully functional robotic agents with real robot platforms, including a robotic arm (AIRBOT Play), a dual-arm humanoid mobile manipulator (AIRBOT MMK2), a wheeled loco-manipulator, a quadrocopter, and other commonly used simulation agents [1], [3]. The diversity of embodiment types enables a comprehensive exploration of various perception and interaction capabilities in DISCOVERSE.

Asset Formatting. For interactive scene nodes, i.e., interactive objects and agents, we utilize a dual 3DGS-Mesh representation, where 3DGS (.ply) generates high-fidelity visuals, and the mesh counterpart (.obj/.stl), further described in the MJCF scene description language, ensures accurate physics simulation. Each interactive object or agent model in DISCOVERSE is linked to a corresponding .xml file, which can be efficiently loaded into the simulation. We either randomize or manually adjust physical properties, such as friction, damping, and density, within appropriate ranges. To ensure accurate contact simulation, we decompose meshes into convex parts [32]. For robot models, Python APIs are provided to assemble the robot piece by piece using URDF. For background nodes, we construct only the native 3DGS representation for rendering, as no physical interaction is required.

C. Sensor

DISCOVERSE enables high-fidelity simulation of diverse sensor modalities, including rendering-based sensors (RGB, depth, LiDAR) and physics-based sensors (contact force, body and joint states, IMU, and optical tactile sensor). Specifically, we utilize Gaussian splatting to enable direct simulation of RGB and depth by alpha-compositing the color and z-position of the intersected Gaussian primitives. We further propose a BVH-accelerated, native Gaussian ray tracing framework to enable highly efficient LiDAR simulation (>100 FPS). We integrate Tacchi [33] for optical tactile simulation and rely on MuJoCo for other physics-based sensors.

D. Actuator

Our actuator model integrates the framework from MuJoCo, incorporating control inputs and optional activation states to represent muscles and pneumatic cylinders with first-order dynamics. The model supports fixed or state-dependent gains, capturing force-length-velocity properties of muscles. Actuators transmit forces to the multi-joint system through direct joint actuation, tendon-driven mechanisms, or slider-crank systems that convert linear motion into angular movement.

E. Throughput Report

Due to its massive parallelization capabilities, DISCOVERSE achieves a total of 650 FPS for rendering hyper-realistic RGB-D frames at 640×480 resolution with 5 cameras on a desktop with Ubuntu 20.04, on 3.1 GHz Intel Xeon w5-3435x CPU and an Nvidia 6000 Ada GPU, and achieves 240 FPS with the same setup on a laptop with Ubuntu 20.04, on 3.2 GHz AMD R7-5800H CPU and an Nvidia GeForce RTX 3060 GPU.

IV. REAL2SIM PIPELINE

We now present the Real2Sim pipeline of DISCOVERSE for high-fidelity asset generation. Our pipeline leverages 3DGS [19] as a universal visual representation and integrates laser scanning, state-of-the-art generative models, and PBR techniques to boost the photorealism of the reconstructed radiance fields.

In this section, we first describe the scene-level Real2Sim approach for generating the background node (Sec. IV-A), followed by an introduction to our object-level Real2Sim approach for generating interactive scene nodes (Sec. IV-B).

A. Background Node Real2Sim Generation

To generate high-fidelity, scene-level assets for use as the background node in simulation, we propose to reconstruct real-world scenes as 3DGS fields with geometry regularization. Additionally, we leverage generative models to recover the environmental lighting, so as to augment the appearance of the interactive nodes for photorealistic rendering.

Real-world Scene Reconstruction. Our goal is to robustly digitize real-world complex scenes across various types and scales. However, optimizing 3DGS fields purely with multi-view images oftentimes leads to suboptimal results, e.g., with excessive blurries and floaters due to the degraded geometry. Therefore, we use LixelKity K1 scanner[1] to obtain reliable geometry measurements to regularize the radiance fields, similar to [34], [35], [36].

Lighting Estimation. Ideally, the interactive scene nodes should be reconstructed independently of the scene and can be seamlessly integrated into any digitized environment (detailed later in Section IV-B). To make the appearance of these nodes better align with the reconstructed scene radiance, we propose to estimate the environmental illumination to simulate realistic distant lighting effects with PBR.

To achieve this, we employ DiffusionLight [37] to generate the HDR environment map from a single image of the scene, which maintains photorealism while enables plausible variations for randomization.

[1]https://xgrids.com/lixelk1

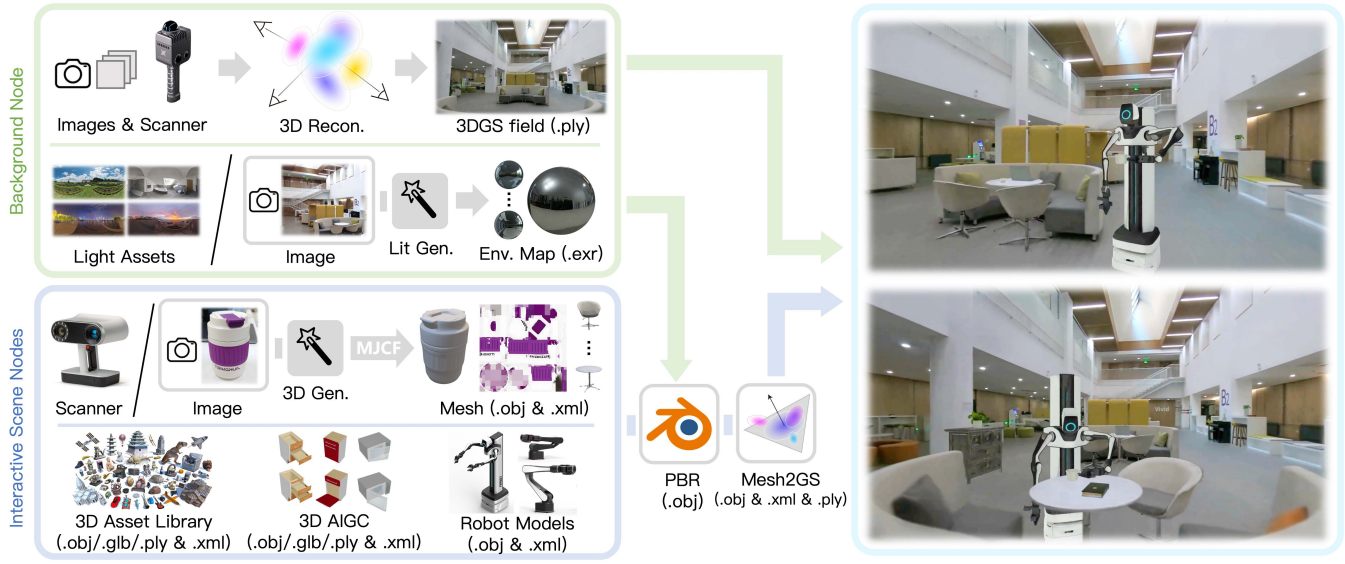

Fig. 3. DISCOVERSE Real2Sim generation pipeline. We use 3DGS as a universal visual representation and integrate laser scanning, state-of-the-art generative models, and physically-based relighting to boost the geometry and appearance fidelity of the reconstructed radiance fields.

B. Interactive Scene Nodes Real2Sim Generation

As described in Sec. III, interactive scene nodes require a dual 3DGS-Mesh representation to support both high-fidelity rendering and accurate physics simulation. To make the best use of off-the-shelf tooling, we propose to first work on the classical textured mesh representation for 3D reconstruction, 3D generation, geometry processing, and Pre-PBR. We then develop an efficient Mesh2GS transfer approach to seamlessly integrate the processed mesh into the splatting renderer.

Laser-scanned 3D Reconstruction. For objects with approximately Lambertian surface, we use Artec Leo[1] scanner to reconstruct the precise 3D geometry and rich textures. Specifically, we place the target object on a turntable and perform multiple scans by rotating and flipping. We then register [38] these scans to obtain the complete textured mesh.

3D Generation. Real-world objects commonly exhibit non-Lambertian specularity or thin structures, posing great challenges for scanners and traditional image-based 3D reconstruction techniques. Fortunately, recent advances in native 3D generation [27], [24] have demonstrated great success in learning useful geometric priors from high-quality 3D data. We thus choose a recent 3D generation model, CLAY[2] [24], to faithfully reconstruct objects with challenging materials or shapes, using a single real-world image as conditional signal.

Relighting. Since it is impractical to capture each object or robot in the scene, the appearance of interactive nodes generally differ a lot from the reconstructed background node due to inconsistent environmental illuminations, resulting in a noticeable gap towards photorealism. To mitigate this issue, we use the estimated HDR environment map (Section IV-A) and Blender[3] to mimic distant lighting effects. To prioritize the rendering throughput of DISCOVERSE, we do not perform the conventional PBR online. Instead, we treat PBR as a preprocessing step to align the appearance of the interactive nodes with the background radiance.

Mesh-Gaussian Transfer. So far, we have obtained high-quality geometry and rendering of interactive scene nodes in the form of textured meshes. However, these are not directly compatible with the 3DGS-based background node. To address this, we propose Mesh-Gaussian transfer, to enable a robust and efficient transition between these two representations.

To convert from mesh to 3DGS, we initialize one Gaussian for each mesh facet and locate the centroid of the Gaussian on the barycenter of the respective triangle. We also flatten 3D Gaussian ellipsoids to 2D slanted planar primitives and align the local frame of each Gaussian with the corresponding face normal. We initialize the size of the Gaussian by setting the scaling values of the tagent axes to be the mean barycenter-to-vertex distance. During optimization, we incorporate additional geometric constraints in terms of depth and opacity to regularize the 3DGS field.

In cases when conversion from 3DGS to mesh is required, we first render multi-view depth maps from the 3DGS field and then apply TSDF fusion [39] and decimation [40].

C. Domain Randomization

To further mitigate domain shifts and account for unmodelled real-world dynamics, we exploit several randomization mechanisms for data augmentation. These include random video overlays (where Internet video crops are randomly selected and linearly blended into the simulation stream), HSV-space image augmentation, and random gamma correction. We also extend the latest generative randomization

[1]https://www.artec3d.com/portable-3d-scanners/artec-leo-fb
[2]https://hyperhuman.deemos.com/rodin
[3]https://www.blender.org/

TABLE II

ZERO-SHOT SIM2REAL SUCCESS RATES OF ACT [20] TRAINED ON DISCOVERSE AND OTHER SIMULATORS.

Tasks	Real2Real	MuJoCo [1]	RoboTwin [6]	SplatSim [18]	**DISCOVERSE**	†MuJoCo [1]	†RoboTwin [6]	†SplatSim [18]	†**DISCOVERSE**
Close-Laptop	100%	2%	0%	56%	**66%**	6%	0%	72%	**86%**
Push-Mouse	94%	48%	24%	68%	**74%**	64%	42%	74%	**90%**
Pick-Up-Kiwifruit	100%	8%	0%	26%	**48%**	36%	0%	44%	**76%**
Average	**98.5%**	14.5%	6%	44%	**55%**	26.5%	10.5%	68%	**86.5%**

† with image-based data augmentation. We perform random video overlays, HSV-space randomization, and random gamma corrections.

TABLE III

ZERO-SHOT SIM2REAL SUCCESS RATES OF DIFFUSION POLICY [21] TRAINED ON DISCOVERSE AND OTHER SIMULATORS.

Tasks	Real2Real	MuJoCo [1]	RoboTwin [6]	SplatSim [18]	**DISCOVERSE**	†MuJoCo [1]	†RoboTwin [6]	†SplatSim [18]	†**DISCOVERSE**
Close-Laptop	100%	0%	0%	70%	**86%**	0%	0%	90%	**96%**
Push-Mouse	96%	0%	22%	54%	**60%**	26%	36%	82%	**88%**
Pick-Up-Kiwifruit	94%	0%	0%	12%	**22%**	6%	0%	52%	**74%**
Average	96.6%	0%	7.3%	45.3%	**56%**	10.6%	12%	74.6%	**86%**

† with image-based data augmentation. We perform random video overlays, HSV-space randomization, and random gamma corrections.

approach [41], leveraging ControlNet [42] for effective conditioning, GPT-4V [43] for text-prompt augmentation, and a hybrid flow-based pipeline [44], [45] for frame interpolation.

V. EXPERIMENTS AND APPLICATIONS

To evaluate the efficacy of DISCOVERSE in bridging the Sim2Real gap, we conduct extensive experiments on imitation learning across three real-world manipulation tasks. We compare the zero-shot Sim2Real success rates of ACT [20] and Diffusion Policy (DP) [21] trained in DISCOVERSE, against those trained in other simulators, including MuJoCo [1], RoboTwin [4], and SplatSim [18], as well as Real2Real transfer. We also report the effects of image-based randomization and outline two exemplar use cases of DISCOVERSE, including navigation and multi-agent coordination, to showcase the versatility of our workflows.

A. Benchmark on Imitation Learning Based Manipulation

Data Collection. Real-world demonstrations are manually collected by a human expert, whereas DISCOVERSE automates this process with motion planners and a gamepad-based state generation approach, which greatly facilitates demonstration by recording the keypoints that describe the relative pose between the robot and object. To collect 100 demonstrations, it takes **146** minutes for real-world collection, but only **1.5** minutes in DISCOVERSE, leading to an $\sim$**100**$\times$ increase in efficiency and demonstrating the scalability of DISCOVERSE.

Evaluation Protocol. We benchmark Sim2Real policy transfer across three contact-rich real-world tasks:

1) *Close-Laptop*: manipulate the lid and close a laptop;
2) *Push-Mouse*: push the mouse onto the mouse pad;
3) *Pick-Up-Kiwifruit*: grasp and pick up a kiwifruit;

We use the publicly available ACT [20] and Diffusion Policy (DP) [21] for imitation learning, and we compare against three state-of-the-art simulators: MuJoCo [1], RoboTwin [6], and SplatSim [18], based on their open-source implementations. For all simulators, we generate 100 demonstrations each task for ACT and 2,000 demonstrations for DP, with random initial gripper states, end-effector positions and orientations. We run 50 trials for each task during testing and use the task success rate (%) as the metric to evaluate Sim2Real and Real2Real performance. All experiments are conducted using an AIRBOT Play robotic arm equipped with an AIRBOT-Gripper-2 and 2 LRCP V1080P cameras (for ego and third-person view), with deployment on an NVIDIA RTX 4090 GPU.

Sim2Real Benchmark. The comparisons on Sim2Real success rates between DISCOVERSE and other simulators are presented in Table II (ACT) and Table III (DP). Policies trained on MuJoCo [1] and RoboTwin [6] struggle with zero-shot Sim2Real transfer for both ACT and DP, primarily due to the significant domain shifts resulting from the poor visual fidelity. While SplatSim [18] partially addresses this issue with 3DGS-based Real2Sim, DISCOVERSE achieves the best Sim2Real results due to superior fidelity, outperforming SplatSim by $\sim$11% on the average task success rate for both ACT and DP, without any data augmentation.

The Sim2Real results with image-based data augmentation are presented in the right sections of Table II and Table III. All simulators show notable improvements after randomization, and DISCOVERSE achieves a 31.5% increase on the average success rate for ACT and a 29.3% improvement for DP. Remarkably, DISCOVERSE outperforms SplatSim by 18.5% for ACT and by 11.4% for DP, when employing the same data augmentation mechanisms.

B. Exemplar Applications

DISCOVERSE is a unified and versatile simulator, with extendable supports for a variety of robot models and downstream tasks. We now showcase two exemplar applications in the context of navigation and multi-agent coordination.

Navigation. We deploy an AIRBOT MMK2 agent to navigate a large-scale indoor scene in DISCOVERSE, following predefined key points. The agent takes ego-view renderings as input and progressively updates the spatial map [23]. Fig. 5 illustrates the visual inputs and the reconstructed spatial map at different timestamps during the exploration.

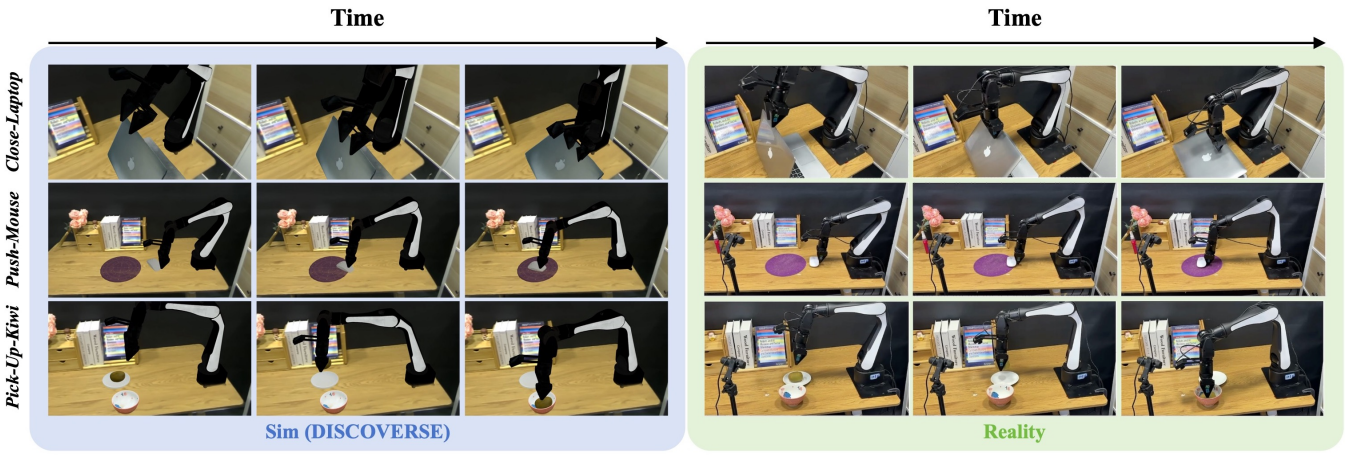

Fig. 4. Visualizations of an AIRBOT Play robotic arm performing three different manipulation tasks in the simulation of DISCOVERSE and in reality.

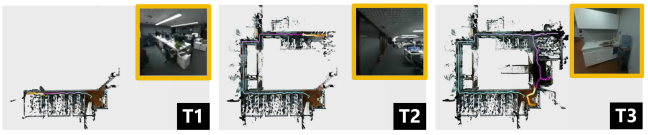

Fig. 5. Visualizations of an agent exploring a large-scale indoor scene in DISCOVERSE at different timestamps. The yellow boxes indicate the ego-view inputs generated by the DISCOVERSE renderer.

Fig. 6. Visualizations of a quadrocopter and a wheeled loco-manipulator cooperatively exploring a large-scale outdoor scene in DISCOVERSE.

Multi-Agent Coordination. As another example shown in Fig. 6, we deploy a wheeled loco-manipulator and a quadrocopter for cooperative exploration of a large-scale in-the-wild scene with DISCOVERSE. The quadrocopter can transform to a wheeled mode, enabling it to land on the wheeled loco-manipulator or the ground for efficient delivery in hard-to-reach areas.

VI. Conclusion and Futher Work

In this work, we introduce DISCOVERSE, the first unified, modular, open-source 3DGS-based simulation framework to bridge the Sim2Real gap in robot learning. By integrating Gaussian Splatting and MuJoCo, DISCOVERSE enables hyper-realistic simulation of complex real-world scenarios, with inclusive supports for various sensor modalities, existing 3D assets, robot models, ROS plugins, and various randomization approaches. These features make DISCOVERSE ideal for large-scale robot learning, efficient data synthesis, and complex robotic benchmarks, e.g., manipulation, navigation, multi-agent coordination, etc. Through extensive experiments on imitation learning with ACT and DP, we verify the superiority of DISCOVERSE in closing the Sim2Real gap, compared to existing simulators.

Future Work. We believe DISCOVERSE represents the beginning of a new era in Real2Sim2Real robot learning, paving the way for end-to-end optimization across the entire Real2Sim and Sim2Real pipeline. While Sim2Real in IL is just the beginning, we envision the capability of DISCOVERSE to facilitate the zero-shot Sim2Real transfer for more complex RL policies beyond IL, thus alleviating the onerous process of real-world RLPF. Future improvements for DISCOVERSE will focus on advanced physical simulation, Mesh-GS hybrid PBR, etc. With DISCOVERSE, we plan to establish a variety of Sim2Real benchmarks for end-to-end robot learning for stimulating further research and practical applications in the field.

VII. Acknowledgement

This work is supported by funding from XIAOMI FOUNDATION. The authors would like to acknowledge DISCOVER Lab and DISCOVER Robotics for technical and hardware supports; Liang Zhu for optimizing the user interface; Shihui Zhou for integrating imitation learning algorithms.

References

[1] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[2] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in neural information processing systems*, vol. 34, pp. 251–266, 2021.

[3] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[4] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 097–11 107.

[5] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber *et al.*, "Threedworld: A platform for interactive multi-modal physical simulation," *arXiv preprint arXiv:2007.04954*, 2020.

[6] Y. Mu, T. Chen, S. Peng, Z. Chen, Z. Gao, Y. Zou, L. Lin, Z. Xie, and P. Luo, "Robotwin: Dual-arm robot benchmark with generative digital twins (early version)," *arXiv preprint arXiv:2409.02920*, 2024.

[7] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[8] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4497–4506.

[9] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain *et al.*, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[10] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE transactions on automation science and engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[11] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.

[12] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[13] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[14] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.

[15] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 713–720, 2020.

[16] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang, "Robogsim: A real2sim2real robotic gaussian splatting simulator," *arXiv preprint arXiv:2411.11839*, 2024.

[17] H. Lou, Y. Liu, Y. Pan, Y. Geng, J. Chen, W. Ma, C. Li, L. Wang, H. Feng, L. Shi *et al.*, "Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation," *arXiv preprint arXiv:2408.14873*, 2024.

[18] M. N. Qureshi, S. Garg, F. Yandún, D. Held, G. Kantor, and A. Silwal, "Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting," *ArXiv*, vol. abs/2409.10161, 2024.

[19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[20] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[21] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[22] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016. [Online]. Available: https://pybullet.org/wordpress/

[23] Y. Li, Z. Kuang, T. Li, G. Zhou, S. Zhang, and Z. Yan, "Activesplat: High-fidelity scene reconstruction through active gaussian splatting," *arXiv preprint arXiv:2410.21955*, 2024.

[24] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu, "Clay: A controllable large-scale generative model for creating high-quality 3d assets," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–20, 2024.

[25] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[26] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 501–518.

[27] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," *arXiv preprint arXiv:2412.01506*, 2024.

[28] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," in *European Conference on Computer Vision*. Springer, 2025, pp. 1–18.

[29] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[30] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 909–918.

[31] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre *et al.*, "Objaverse-xl: A universe of 10m+ 3d objects," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[32] X. Wei, M. Liu, Z. Ling, and H. Su, "Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–18, 2022.

[33] Z. Chen, S. Zhang, S. Luo, F. Sun, and B. Fang, "Tacchi: A pluggable and low computational cost elastomer deformation simulator for optical tactile sensors," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1239–1246, 2023.

[34] J. Cui, J. Cao, Y. Zhong, L. Wang, F. Zhao, P. Wang, Y. Chen, Z. He, L. Xu, Y. Shi *et al.*, "Letsgo: Large-scale garage modeling and rendering via lidar-assisted gaussian primitives," *arXiv preprint arXiv:2404.09748*, 2024.

[35] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.

[36] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.

[37] P. Phongthawee, W. Chinchuthakun, N. Sinsunthithet, V. Jampani, A. Raj, P. Khungurn, and S. Suwajanakorn, "Diffusionlight: Light probes for free by painting a chrome ball," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 98–108.

[38] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[39] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[40] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 209–216.

[41] A. Yu, G. Yang, R. Choi, Y. Ravan, J. Leonard, and P. Isola, "Learning visual parkour from generated images," in *8th Annual Conference on Robot Learning*, 2024.

[42] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3836–3847.

[43] "Gpt-4v(ision) system card," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263218031

[44] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.

[45] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer, 2003, pp. 363–370.