Article | Received 11 December 2025; Revised 22 February 2026; Accepted 6 April 2026; Published 13 May 2026
https://doi.org/10.55092/rl20260013

A high-fidelity digital twin for robotic manipulation based on 3D Gaussian Splatting

Ziyang Sun, Lingfan Bao, Tianhu Peng, Jingcheng Sun and Chengxu Zhou*

Department of Computer Science, University College London, London, UK

* Correspondence author; E-mail: chengxu.zhou@ucl.ac.uk.

Highlights:

- Proposes a unified perception-to-planning-to-execution workflow leveraging 3DGS to create photorealistic digital twins from sparse RGB views in minutes, forming a holistic, closed-loop pipeline from capture to real-robot execution.
- Introduces a robust method for semantic understanding by lifting 2D masks from foundation models like SAM into the 3D scene using multi-view spatial consensus, and an efficient conversion of raw 3DGS into planning-ready collision geometry for physics-based simulation.
- Demonstrates real-world effectiveness with physical experiments on a Franka Emika Panda robot, showing that motion plans validated in the digital twin can be transferred to real-robot execution.

Abstract: Developing high-fidelity, interactive digital twins is crucial for enabling closed-loop motion planning and reliable real-world robot execution, which are essential to advancing sim-to-real transfer. However, existing approaches often suffer from slow reconstruction, limited visual fidelity, and difficulties in converting photorealistic models into planning-ready collision geometry. We present a practical framework that constructs high-quality digital twins within minutes from sparse red, green, blue (RGB) inputs. Our system employs 3D Gaussian Splatting (3DGS) for fast, photorealistic reconstruction as a unified scene representation. We enhance 3DGS with visibility-aware semantic fusion for accurate 3D labelling and introduce an efficient, filter-based geometry conversion method to produce collision-ready models seamlessly integrated with a Unity-ROS2-MoveIt physics engine. In experiments with a Franka Emika Panda robot performing pick-and-place tasks, we demonstrate that this enhanced geometric accuracy effectively supports robust manipulation in real-world trials. These results demonstrate that 3DGS-based digital twins, enriched with semantic and geometric consistency, offer a fast, reliable, and scalable path from perception to manipulation in unstructured environments.

Keywords: 3D Gaussian Splatting; digital twin; robotic manipulation; Real-to-Sim-to-Real

1. Introduction

The field of robotics is rapidly moving towards full autonomy in complex and unstructured environments. Effective autonomous manipulation in unstructured environments fundamentally relies on the robot's ability to rapidly construct a high-fidelity, actionable understanding of its surroundings, which is a core requirement for achieving advanced tasks such as fine-grained manipulation. This actionable understanding relies heavily on the construction of an accurate virtual replica, commonly known as a digital twin [1]. Digital twins are essential tools that enable safe, repeatable validation, closed-loop motion planning, and reliable sim-to-real transfer, which is important for advancing the deployment of robotic systems in the real world.

However, existing reconstruction pipelines [2] introduce significant bottlenecks that impede their seamless integration into real-time robotic workflows. The pursuit of visual fidelity often conflicts with the need for computational efficiency and physical utility. Specifically, Neural Radiance Fields (NeRF) [3,4] offer high photorealism but are computationally expensive, often requiring minutes to hours of time for optimisation, which severely limits rapid deployment. Conversely, traditional methods based on point clouds or mesh reconstruction [5] are faster but often suffer from insufficient fidelity, noise sensitivity, and the difficulty of projecting reliable, consistent semantic labels from sparse multi-view inputs.

In this context, 3D Gaussian Splatting (3DGS) [6], a novel explicit radiance field reconstruction method, has shown outstanding performance in both reconstruction quality and speed and has emerged as a potential breakthrough representation. 3DGS successfully balances the trade-off between speed and fidelity, achieving photorealistic rendering quality within minutes, which is highly promising for rapid robotic scene capture. However, the core representation used by 3DGS, anisotropic Gaussian splats (or "balls"), defines each scene primitive not merely by a point but by a position, a covariance matrix, a colour, and an opacity. Although this representation excels at combining colour and alpha (opacity) components to create a visually convincing and continuous surface view, the underlying explicit geometry remains inherently ambiguous and problematic for physical interaction. Specifically, the resulting cloud of stretched Gaussian primitives is not a clean, watertight surface, but is riddled with reconstruction artefacts. These issues include floaters (isolated clusters derived from optimisation residuals), ghost artefacts (semi-transparent, low-opacity points near reflective or occluded boundaries), and overall surface fuzziness. These geometric imperfections, while visually hidden in the rendered image, render the raw 3DGS output unsuitable for precise robotic tasks such as collision checking and motion planning.

The fragmentation between these approaches highlights a critical, unsolved challenge: generating a digital twin that is simultaneously photorealistic, rapidly reconstructed, and equipped with planning-ready collision geometry and consistent semantic structures. Addressing this triple constraint, speed, fidelity, and actionability, is the central challenge for next-generation robotic perception systems.

To bridge the gap between visual fidelity, reconstruction efficiency, and physical utility in digital twin construction, we present a unified framework that generates interactive, semantically structured digital twins directly from sparse red, green, blue (RGB) inputs.

To ensure the resulting environment is geometrically interactive and planning-ready, we introduce two critical components: (i) a visibility-aware semantic fusion module that aggregates multi-view

cues [7,8] to achieve consistent 3D semantic labelling of the Gaussian primitives, and (ii) a geometric refinement process that addresses the inherent limitations of Gaussian-based representations—specifically the presence of floater artefacts and noisy density distributions—by converting these primitives into precise, collision-ready meshes. This conversion is vital for transforming a purely visual reconstruction into a physically actionable asset that meets the strict geometric requirements of motion planners.

While recent initiatives like safe real-time robot navigation in Gaussian Splatting maps (Splat-Nav) [9] and splat-multi-stage, open-vocabulary robotic manipulation via editable Gaussian Splatting (Splat-MOVER) [10] have integrated 3DGS into robotic workflows, they predominantly target mobile navigation or semantic affordance detection, often relying on coarse occupancy proxies that lack the geometric fidelity required for fine-grained manipulation. Similarly, Real2Sim2Real robotic Gaussian Splatting simulator (RoboGSim) [11] represents a significant step forward by utilising 3DGS to construct photorealistic environments for offline reinforcement learning. However, its primary focus lies in bridging the visual domain gap for policy training rather than enabling rapid, online geometric perception. In contrast, our framework prioritises immediate physical actionability by focusing on rapid planning-ready reconstruction, converting image-level input into collision-ready meshes within minutes to support precise, solver-based motion planning in the real world.

Finally, we deploy these assets into a custom Unity-based environment to facilitate a validated Real-to-Sim-to-Real workflow, enabling motion plans generated within the high-fidelity digital twin to be validated before transfer to physical robot execution. This closed-loop pipeline, integrating photorealistic reconstruction, semantic understanding, collision geometry, and validated simulation execution, represents progress toward fully actionable digital twins for complex robotic manipulation.

The main contributions of this work are summarised as follows:

(1) Unified Real-to-Sim-to-Real Framework: We propose a unified Real-to-Sim-to-Real framework that synergises 3DGS with robust point cloud processing. This approach generates actionable digital twins within minutes, effectively bridging the gap between neural rendering and robotic manipulation.

(2) Visibility-Aware Semantic Fusion: We introduce a view-dependent semantic aggregation with occlusion-aware confidence weighting strategy. This method distils 2D segmentation cues from vision foundation models [7,12] into consistent 3D attributes, resolving projection ambiguities to ensure accurate labelling under occlusion and providing a semantically structured environment for robot manipulation.

(3) Planning-Ready Geometry Conversion: Addressing the limitation of 3DGS as a purely visual representation, we implement a multi-stage geometric refinement process with attribute-based pruning, connectivity analysis, and alpha-shape meshing. By combining opacity-based thresholding, scale-based filtering, and density-based spatial clustering of applications with noise (DBSCAN) clustering before mesh generation, this step converts raw Gaussian splat point clouds into planning-ready collision meshes.

(4) Real-World Closed-Loop Validation: Beyond simulation-only validation, we demonstrate a complete perception-planning-validation-execution loop on a Franka Emika robot [13], showing that plans generated from the converted collision geometry transfer to physical execution.

The remainder of this paper is organised as follows. Section 2 reviews related work on 3D reconstruction,

semantic understanding, and digital twins for manipulation. Section 3 presents the proposed methodology, including high-fidelity reconstruction, visibility-aware semantic fusion, physics-ready geometry reconstruction, and system integration. Section 4 describes the experimental setup, evaluation metrics, and experimental results. Section 5 discusses the limitations of the current framework. Section 6 concludes the paper and outlines future work.

2. Related work

2.1. 3D scene reconstruction for robotics

While dense mapping pipelines like truncated signed distance function (TSDF) [14] and Voxblox [15] have long served as the backbone for robotic navigation, their dependence on voxel discretisation fundamentally limits their utility for manipulation. The resulting over-smoothed geometry fails to capture the high-frequency surface details necessary for fine motor control. Similarly, while implicit representations such as neural radiance fields (NeRF) [3] and instant-neural graphics primitives (Instant-NGP) [16] offer visual photorealism, they remain constrained by prohibitive inference latencies and dense view requirements, creating a bottleneck for real-time robotic exploration.

Recently, 3DGS [6] has emerged to bridge this gap by explicitly representing scenes as anisotropic Gaussian primitives, combining differentiable optimisation with fast rasterisation, offering photorealistic rendering at real-time speeds. However, despite this visual fidelity, raw 3DGS representations are inherently ill-suited for physical interaction. The presence of reconstruction artefacts, such as floaters, ghost points, and surface fuzziness, makes the resulting point clouds functionally unsuitable for direct collision checking and robot manipulation.

In contrast, our framework is specifically designed to close the gap between visual realism and physical validity. By systematically structuring raw 3DGS outputs, we transform noisy visual primitives into manipulation-ready geometry without sacrificing the rendering speed required for online operation.

2.2. Semantic scene understanding

Robotic manipulation requires a precise understanding of object identity beyond mere geometry. Although foundation models like segment anything model (SAM) [7] and Grounded SAM [8] have revolutionised 2D segmentation, lifting these predictions into 3D space remains a critical challenge. Existing feature distillation methods, such as SegmentAnyGaussian [17], attempt to solve this by appending high-dimensional vectors to primitives, but this approach drastically increases memory footprints and training overhead, limiting deployment agility. Conversely, direct projection methods often suffer from “bleeding” labels and inconsistencies caused by depth discontinuities.

What sets our approach apart is that it diverges from these computationally heavy or unstable methods by introducing a Visibility-Aware Semantic Fusion module. Instead of relying on extensive retraining or naive projection, our pipeline is grounded in a rigorous geometric consensus mechanism. This method integrates depth and visibility checks with a confidence-weighted voting scheme, ensuring that semantic labels are not just projected, but geometrically verified across views. This foundation enables our system to achieve high-fidelity 3D labelling that is both consistent and computationally lightweight.

2.3. *Digital twins for interactive manipulation*

The ultimate goal of robotic perception is to enable interaction. Ideally, a digital twin must unify three capabilities: photorealistic rendering, semantic understanding, and physical collision handling. Traditional simulators (e.g., Gazebo, MuJoCo) achieve physics but lack visual fidelity, while neural renderers achieve fidelity but lack physical structure. Bridging this "Sim-to-Real" gap requires a hybrid representation.

Recent efforts have begun to integrate 3DGS into planning frameworks. Splat-Nav [9] utilises Gaussian representations for navigation, employing ellipsoid abstractions for collision checking. However, its focus is global path planning rather than the object-level granularity required for grasping. Similarly, RoboGSim [11] focuses on the simulation aspect, providing a platform for offline testing rather than an online perception-to-action pipeline. Other works like Splat-MOVER [10] and GraspSplats [18] explore open-vocabulary manipulation and 3D feature splatting for grasping, respectively.

Despite these advancements, existing frameworks face fundamental gaps regarding manipulation. Firstly, the presented works do not provide robust mechanisms to convert raw, noisy 3DGS outputs (often plagued by floaters and ghost artefacts) into planning-ready collision geometry. Secondly, existing works primarily address visual policy learning or navigation, leaving the validation of classical motion planning on reconstructed geometry unexplored. In particular, RoboGSim [11] provides a valuable 3DGS-enabled simulation platform for robot learning. However, it primarily utilises 3DGS for photorealistic rendering rather than as a source for collision geometry generation. The systematic conversion of Gaussian primitives into planning-compatible collision geometry, and its validation through real-robot execution, remain underexplored. Our work directly addresses both challenges. Addressing these limitations requires a unified pipeline that integrates sparse-view reconstruction [19], geometrically-verified semantic consensus, and physics-based post-processing [20,21] into a reliable robotic workflow.

Our proposed framework addresses this fragmentation by establishing a unified closed-loop Real-to-Sim-to-Real pipeline, as shown in Figure 1. It seamlessly integrates photorealistic 3DGS reconstruction with our refined collision geometry (using alpha-shape meshing), semantic understanding, and a Unity-based robot simulation and manipulation interface [22,23]. This complete, validated sim-to-real workflow is a key advancement, ensuring that motion plans generated against the high-fidelity digital twin are reliable for real-world execution on the real robot.

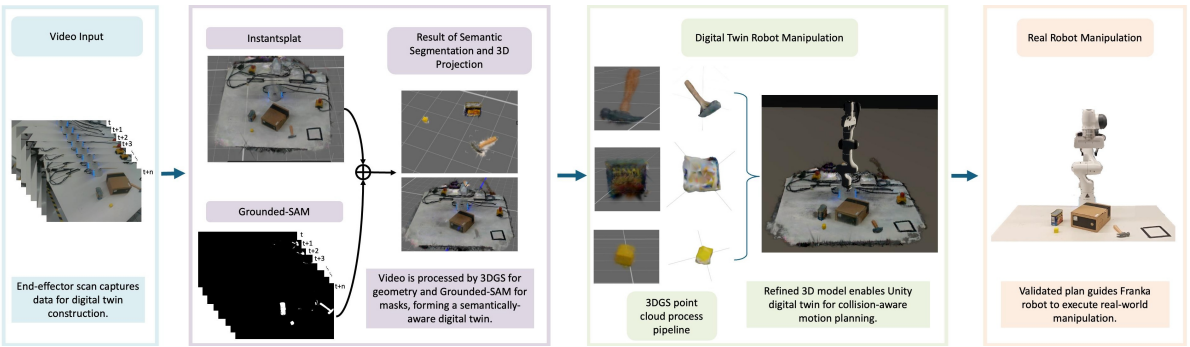


Figure 1. The overall pipeline of this framework uses multi-view video input and 3DGS to reconstruct the scene geometry. Grounded SAM provides semantic masks, which are fused with the 3D projection to form a semantically aware digital twin. This twin enables collision-aware motion planning for real robot manipulation.

3. Methodology

Our proposed framework establishes a comprehensive digital twin generation pipeline tailored for closed-loop robotic manipulation. Following the framework pipeline illustrated in Figure 1, the system operates via two parallel processing streams: (1) a geometric reconstruction stream that uses an optimised 3DGS approach to generate a high-fidelity 3D scene, and (2) a semantic segmentation stream that identifies and isolates manipulable objects. The subsequent stages focus on rigorously transforming this semantically-annotated 3D model into clean, planning-ready collision geometry and integrating it into the simulation environment for validation.

3.1. High-fidelity scene reconstruction

We employ a 3DGS-based approach for scene reconstruction due to its superior balance of rendering quality and rapid optimisation speed. This choice is important for achieving fast reconstruction and high-quality digital twins, contrasting sharply with NeRF-based methods whose latest algorithms commonly require tens of minutes or more for comparable fidelity. To address the challenges posed by sparse and uncalibrated input images, which often lead to failures in traditional Structure-from-Motion (SfM) pipelines, we use the InstantSplat methodology [19]. This streamlined approach eliminates the need for a separate SfM step by utilising a pre-trained geometric prior, such as matching and stereo 3D reconstruction (MASt3R) [24], to directly estimate an initial point cloud and camera poses.

The core of the reconstruction is a self-supervised optimisation process. The scene is represented by a set of 3D Gaussians, each defined by a position $\boldsymbol{\mu}$, a covariance matrix $\boldsymbol{\Sigma}$, a colour, and an opacity. The unnormalised density is given by [6]:

$$G(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{1}$$

This set of Gaussians is jointly optimised with the camera poses to minimise the photometric rendering error between the rendered images and the input views. This technique bypasses the traditional, time-consuming adaptive density control steps of vanilla 3DGS, enabling extremely fast convergence and yielding a high-fidelity 3D representation suitable for photorealistic rendering and depth extraction.

3.2. Visibility-aware semantic fusion

Achieving a reliable, actionable semantic understanding is the prerequisite for robot interaction. However, a fundamental conflict arises when lifting 2D perception to 3D: standard single-view segmentation models (e.g., SAM) suffer from the "bleeding effect," where background pixels near the object boundary are erroneously included in the foreground mask. To resolve this, we propose a visibility-aware semantic fusion framework. Unlike naive projection methods, our approach treats 2D masks as noisy spatial hypotheses and enforces 3D geometric consistency to filter out segmentation outliers.

Spatial Isolation via Depth Clustering: The core premise of our method is that semantic coherence implies spatial coherence. While a 2D segmentation mask M_j may loosely cover both the target object and the adjacent background, the underlying 3D geometry exhibits a distinct depth discontinuity.

To exploit this, we perform Depth-Guided Isolation for each view. We project the 3D Gaussians

contained within the 2D mask and apply DBSCAN on their depth values. We assume the largest cluster corresponds to the true object geometry, while smaller, spatially detached clusters represent background artefacts included by the 2D model.

Confidence-Weighted Consensus: To aggregate these observations into a unified 3D semantic field, we employ a weighted voting mechanism governed exclusively by spatial validity. We define the fusion weight $W_{i,j}$ for Gaussian i in view j as:

$$W_{i,j} = w_{\text{cluster}}(i,j) \tag{2}$$

Here, w_{cluster} acts as a *soft spatial gate*. Points belonging to the primary depth cluster are assigned high confidence ($w_{\text{cluster}} \approx 1$), while spatial outliers are suppressed ($w_{\text{cluster}} \approx 0$). This formulation is robust against the geometric ambiguities of the 2D masks, ensuring that only points physically co-located with the object contribute to the semantic label.

The final semantic label for a 3D point $\mathbf{p}_i$ is determined by accumulating these spatially-weighted votes across all visible views V_i':

$$\mathbf{p}_i \in \begin{cases} \mathscr{P}_{\text{obj}} & \text{if } \sum_{j \in V_i'} W_{i,j} \cdot \mathbb{I}((u_i, v_i) \in M_j) \geq \tau_{\text{consensus}} \\ \mathscr{P}_{\text{bg}} & \text{otherwise} \end{cases} \tag{3}$$

Notation: $\mathbf{p}_i$ denotes the 3D position of Gaussian i (or its mean $\boldsymbol{\mu}_i$). (u_i, v_i) denotes the 2D projection of $\mathbf{p}_i$ into view j under the estimated camera model. M_j is the 2D foreground mask in view j. $\mathbb{I}(\cdot)$ is an indicator function that equals 1 if the predicate is true. V_i' is the set of views where Gaussian i is visible after the depth/visibility test. $W_{i,j}$ is the spatial validity weight defined in Equation (2). $\tau_{\text{consensus}}$ is the consensus threshold controlling the completeness/artefact trade-off.

Table 1 summarises the main notations used in the semantic lifting formulation. These symbols define the 3D point representation, image projection, mask observation, visibility filtering, spatial weighting, and multi-view consensus threshold used in our semantic reconstruction pipeline.

Table 1. Main notations used in semantic lifting.

Symbol	Meaning
$\mathbf{p}_i$	3D point (Gaussian mean) of primitive i
(u_i, v_i)	Pixel coordinate of $\mathbf{p}_i$ projected to view j
M_j	2D segmentation mask in view j
V_i'	Visible-view set for primitive i after depth/visibility checks
$W_{i,j}$	Spatial validity weight for primitive i in view j
$\tau_{\text{consensus}}$	Multi-view vote threshold
N	Number of visible views, $N = \lvert V_i' \rvert$
κ	Strictness factor for the consensus threshold

In practice, we set $\tau_{\text{consensus}} = N/\kappa$. This setting makes the vote requirement increase with view coverage while remaining tolerant to occasional segmentation failures. Based on the consistency-artefact trade-off analysed in Section 4.4.1, we use $\kappa = 1.5$ as the default value.

For DBSCAN-based depth clustering, we empirically determine hyper-parameters to balance object connectivity and artefact rejection. We set $\varepsilon_{\text{depth}} = 0.02\,\text{m}$, which corresponds to approximately 2% to 3% of the typical workspace depth range (0.3 to 1.0 m). This threshold effectively groups Gaussian primitives belonging to the same physical surface (typically < 2 cm apart in depth) while rejecting background floaters and ghosting artefacts that are spatially detached (> 5 cm separation). The minimum cluster size is set to min_samples = 10 to filter out spurious small clusters arising from segmentation noise, a value validated across our object set ranging from 5 to 30 cm in size. These parameters proved robust across all test scenarios. However, deployment at substantially different scales (e.g., warehouse logistics or micro-manipulation) may require recalibration, as discussed in the Limitations section.

This consensus mechanism effectively "carves" the correct semantic shape out of the noisy 2D predictions.

Iterative Semantic Refinement: To further sharpen the boundaries, we implement an iterative feedback loop (max_iter = 3). As the 3D semantic model improves, it generates cleaner depth maps for the next iteration's visibility checks. We incorporate a Boundary Refinement step using K-Nearest Neighbors (KNN) to smooth local label inconsistencies, ensuring continuous surface semantics.

3.3. Physics-ready geometry reconstruction

Following semantic fusion, the scene is partitioned into the target object $\mathscr{P}_{\text{obj}}$ and the environmental background $\mathscr{P}_{\text{bg}}$. However, semantic identity does not guarantee geometric utility. The raw 3DGS representation contains low-density visual artefacts, including floaters, semi-transparent ghost primitives, and stretched needle-like splats, which creates false obstacles for the physics engine.

To resolve this problem, we implement a three-stage reconstruction pipeline applied identically to both the object and background point clouds. This process systematically converts the noisy visual representation into a clean, collision-free physical environment.

3.3.1. Stage 1: intrinsic attribute filtering

The initial phase acts as a global statistical cleaner, removing primitives that contribute to visual haze but lack physical substance. We apply two rigorous filters to the entire scene:

- Opacity Threshold: We discard primitives with low opacity (e.g., $\alpha < 0.1$). This effectively eliminates the semi-transparent "mist" often found hovering above surfaces.
- Geometric Regularisation: We analyse the covariance scales to identify and remove overly stretched, "needle-like" primitives. These artefacts, common in sparse-view areas, are pruned to prevent the physics engine from registering false collisions with non-existent spikes.

3.3.2. Stage 2: semantic-guided connectivity pruning

Even after statistical filtering, isolated clusters of noise (floaters) may persist. We leverage the semantic prior established in the previous section to perform topological cleaning. For any given semantic partition (whether object or background), we assume the physical entity corresponds to the dominant geometric structure.

We employ DBSCAN clustering to segment the point cloud into spatially disjoint groups. By retaining only the largest connected cluster and pruning all smaller detached components, we effectively wipe out

floating artefacts. This ensures that $\mathscr{P}_{\text{obj}}$ resolves to a single coherent object and $\mathscr{P}_{\text{bg}}$ resolves to a clean static environment (e.g., the table surface), free from phantom obstacles.

3.3.3. Stage 3: collision-ready meshing via alpha shapes

Finally, to bridge the gap to robotic manipulation, we convert the cleaned point geometry into a mesh. We select the alpha shapes algorithm, which functions as a "shrink-wrap" operation. Unlike implicit smoothing methods, alpha shapes tightly conform to the point distribution, preserving sharp geometric features, such as box corners and handle edges, that are critical for stable grasping contact.

3.4. Interactive digital twin and planning

The semantically-segmented and geometrically refined 3D models are imported into the Unity engine. The background point cloud $\mathscr{P}_{\text{bg}}$ forms the static environment. Each manipulable object in $\mathscr{P}_{\text{obj}}$ is assigned a MeshCollider for accurate collision detection and a Rigidbody for realistic physics-based interactions. The Unity environment acts as the digital twin.

We establish a seamless, high-bandwidth communication bridge between Unity and the standard robotic software stack (ROS 2) using the ROS2 for Unity plug-in. This enables bidirectional state synchronisation: the robot's state is sent to Unity, and the digital twin's environment geometry and object poses (derived from the segmented point clouds) are dynamically sent to the MoveIt 2 [13] planning scene. The system uses this complete and rapidly generated information to perform collision-aware motion planning. Generated trajectories are first validated in the physics-enabled simulation before being sent to the physical robot for execution, forming a reliable perception-to-planning-to-validation sim-to-real workflow.

It is worth noting that the scope of this work is currently limited to static scenes. This design choice prioritises reconstruction efficiency and geometric stability, key requirements for the proposed rapid scan-and-plan workflow, over the computational complexity associated with dynamic modelling. Furthermore, the static assumption remains valid for the targeted tabletop rearrangement tasks, where the environment is assumed to remain stable during the planning phase.

4. Experiments and results

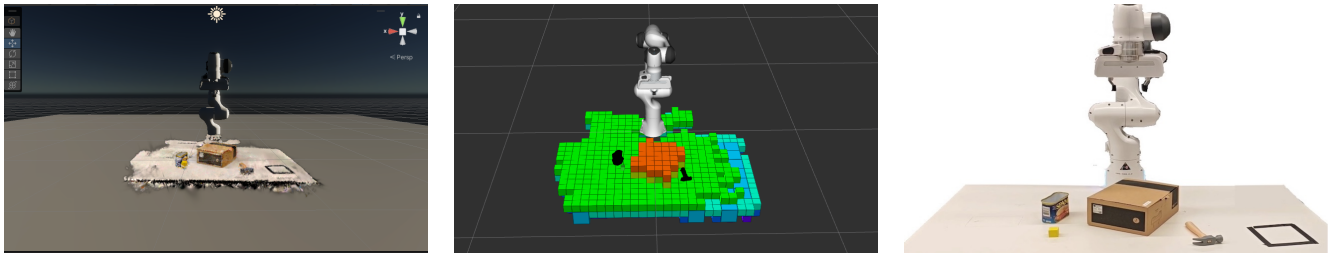
To comprehensively validate the effectiveness and robustness of our proposed high-fidelity 3DGS digital twin framework for robotic manipulation tasks, we designed and conducted a series of quantitative experiments. This section details the experimental setup, task definition, baselines for comparison, and quantitative analysis of key evaluation metrics.

4.1. Experimental setup

Our experimental platform centres on a Franka Emika 7-DOF robot arm equipped with an Intel RealSense D435i RGB-D camera mounted on its end-effector. The camera provides high-resolution colour images that serve as input for our 3DGS framework. All computations were performed on a workstation equipped with an NVIDIA RTX 4090 GPU, ensuring rapid 3DGS training and rendering to meet the demanding requirements for reconstruction efficiency.

To systematically evaluate reconstruction robustness across varying geometric and optical challenges, we selected seven representative objects spanning three difficulty levels. The L1-Basic category includes a Blue Box and Yellow Cube, featuring convex geometry with Lambertian surfaces that serve as baseline objects. The L2-Complex category comprises a Toy Hammer and Scissors, presenting non-convex shapes with thin structures that challenge geometric reconstruction. The L3-Textured category contains a Diet Coke Bottle, Glue Stick, and Pen, exhibiting high-frequency surface details that test the framework's ability to capture fine visual features. These objects enable targeted analysis of reconstruction quality, semantic segmentation accuracy, and geometric fidelity across distinct challenge categories.

As illustrated in Figure 2, we constructed a challenging, unstructured tabletop scene to test the system's zero-shot generalisation capabilities. The scene includes objects varying in geometry, texture, and function: a Toy Hammer with complex geometric shape, a simple-surfaced Blue Box, and a small Yellow Cube. Additionally, a cardboard box serves dual roles, acting as a static obstacle initially and subsequently becoming a target placement area. This dynamic role assignment tests the system's adaptability to environmental changes.



(a) (b) (c)

Figure 2. Integration and validation of the digital twin framework across simulation and reality. The Unity view **(a)** shows the high-fidelity, photorealistic digital twin built with 3DGS and integrated with the physics engine. This model generates and validates collision-aware motion plans visualised in **(b)** the RViz interface, which uses simplified geometry for MoveIt planning. The validated plan is then executed by **(c)** the real Franka Emika robot, completing the sim-to-real workflow.

4.1.1. Task definition

The core evaluation task is defined as a long-horizon, zero-shot rearrangement comprising three sequential manipulation steps. First, in the object-obstacle interaction phase, the robot grasps the Blue Box and places it atop the cardboard box, testing planning and manipulation capabilities in the presence of obstacles. Second, the object-object interaction phase requires grasping the Yellow Cube and placing it on the Blue Box, demanding accurate perception and interaction with previously moved objects. Third, the irregular object manipulation phase involves grasping the geometrically complex Toy Hammer and placing it within a designated target frame, evaluating robustness in reconstructing and manipulating irregular shapes. Successful execution requires proactive planning in a dynamic environment rather than purely reactive perception-based control. Since all objects and scene layouts were unseen during system development, this constitutes a zero-shot manipulation problem.

4.1.2. Comparison methods

To quantitatively evaluate our approach, we compared against two representative baselines focusing on 3D reconstruction fidelity and efficiency. Our 3DGS-based method performs a single scene scan using sparse multi-view RGB images (10 to 20 views). This view count was experimentally determined to represent the optimal trade-off for the "scan-and-plan" workflow: it provides sufficient parallax for robust reconstruction while keeping the robotic data acquisition time within a practical minimum. This allows us to rapidly build a high-fidelity digital twin in minutes, which is then imported into the Unity physics engine for complete planning and pre-validation.

Baseline 1 employs traditional point cloud reconstruction using the Intel RealSense D435i to perform multi-view depth fusion, establishing a benchmark for reconstruction efficiency and geometric accuracy. Baseline 2 utilises a state-of-the-art NeRF framework (Instant-NGP) for 3D scene reconstruction from identical sparse multi-view RGB images, enabling a direct comparison of efficiency and photorealistic quality between 3DGS and NeRF approaches in this sparse-data regime.

4.1.3. Evaluation metrics

Our evaluation encompasses five categories of metrics. For reconstruction fidelity and efficiency, we measure reconstruction time from image capture to model completion, along with Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) computed on held-out novel views.

To assess semantic segmentation accuracy, we evaluate mean Intersection over Union (mIoU) against manually annotated ground truth, alongside two custom consistency metrics. First, we define the 3D Projection Consistency score. Let $\mathcal{V}_p$ be the set of views where a 3D point $\mathbf{p}$ projects strictly within the image sensor boundaries. We denote the number of valid observations as $N_p = |\mathcal{V}_p|$. Within these valid views, let N_p^{fg} be the count of views where the projected pixel falls within the foreground mask region. The consistency score is defined as $\text{Consistency}(\mathbf{p}) = N_p^{\text{fg}}/N_p$. We report the dataset-level consistency as the percentage of valid points (where $N_p > 0$) that achieve a high confidence of $\text{Consistency}(\mathbf{p}) \geq 0.8$. Second, the Ghost Index measures artefact introduction in background regions, defined as the percentage increase in foreground points when relaxing the voting threshold, normalised by the total point count.

Geometric fidelity is evaluated using Chamfer Distance, Precision, and F1-Score against manually cleaned ground truth models. Finally, manipulation task performance is measured through task success rate, collision count, and qualitative placement correctness with respect to the designated goal regions.

4.2. Reconstruction performance

The quantitative results presented in Table 2 demonstrate the superior performance of our proposed 3DGS-based method in both efficiency and rendering quality. Our method achieves an average reconstruction time of 229 seconds across test scenes with varying input sparsity (10 to 20 views, 3000 iterations), representing a 5× speed-up over the NeRF baseline (1123 seconds) and effectively reducing the reconstruction bottleneck from hours to minutes suitable for rapid deployment. Reconstruction time scales with scene complexity, ranging from 109 seconds for simple textured objects to 349 seconds for scenes with complex geometry or view-dependent effects.

Regarding rendering quality, our method achieves an average PSNR of 37.03 dB and SSIM of 0.9821,

representing an 8 dB improvement over NeRF (28.33 dB / 0.9037). Performance varies predictably with object category: well-textured objects achieve high quality (PSNR 41.34 dB, SSIM 0.9934) with sparse inputs, preserving fine details such as legible text and sharp edges, while geometrically simpler objects with uniform surfaces yield intermediate performance around 36.77 dB. Although traditional point cloud approaches offer faster processing (approximately 20 seconds), they lack the photorealistic textures essential for high-fidelity digital twins and suffer from significant drift on consumer-grade depth sensors, rendering them unsuitable for our application.

Table 2. Reconstruction performance comparison. Results averaged across test scenes with varying complexity (10 to 20 input views, 3000 iterations). Times measured on NVIDIA RTX 4090. PSNR/SSIM computed on 5 held-out novel views.

Method	**Time (s)** ↓	**PSNR (dB)** ↑	**SSIM** ↑
Point Cloud	∼20	N/A	N/A
NeRF (Instant-NGP)	1123 ± 200	28.33 ± 1.0	0.9037 ± 0.023
Ours (3DGS)	**229 ± 120**	**37.03 ± 5.0**	**0.9821 ± 0.011**

4.3. Scalability and engineering cost

To assess engineering applicability, we analyse runtime and memory usage as a function of (i) number of input views, (ii) scene size measured by the number of Gaussians after optimisation, and (iii) number of segmented objects. We summarise the observed trends below, with per-configuration measurements reported in Table 3.

Table 3. Scalability of the pipeline with respect to input views and scene complexity. All times measured on an NVIDIA RTX 4090. Peak GPU memory recorded during 3DGS optimisation. #Gaussians counts reflect scene complexity after optimisation.

Views	**#Gaussians (K)**	**3DGS time (s)**	**Geo. conv. (s)**	**Peak GPU (GB)**
10	∼1300	109 ± 18	< 5	18.3 ± 0.8
15	∼1700	229 ± 52	< 5	20.1 ± 1.0
20	∼2100	349 ± 61	< 5	22.4 ± 1.2

Empirically, 3DGS optimisation time scales approximately linearly with the number of input views, ranging from approximately 109 s at 10 views to 349 s at 20 views on the RTX 4090. The geometry conversion stage (filtering, DBSCAN clustering, and alpha-shapes meshing) adds less than 10% overhead relative to the 3DGS training time and scales primarily with the number of retained Gaussians rather than the number of input views. Semantic lifting cost grows linearly with the product of the number of views and the number of target objects, as each object requires per-view mask projection and depth clustering. Peak GPU memory during optimisation remains practical for high-end workstations; however, deployment on embedded platforms may require model pruning or lower-resolution training, as discussed in the Limitations section.

4.4. Semantic segmentation accuracy

4.4.1. Multi-view consistency analysis

A core challenge in lifting 2D masks to 3D is balancing completeness with noise suppression. Table 4 presents the trade-off between consistency score and ghost index under different voting thresholds, where N denotes the number of visible views. A loose threshold ($N/2.0$) achieves low artefact rate (ghost index 22.48%) but suffers from incomplete segmentation (82.41% consistency), often missing object boundaries. Conversely, a strict threshold ($N/1.0$) guarantees 100% consistency but introduces excessive noise (ghost index 67.23%), generating floating obstacles that interfere with motion planning. We selected $N/1.5$ as the operating point, achieving 93.72% consistency while maintaining the ghost index below 50%, ensuring robust 3D object definition without compromising the free space required for collision-free planning.

Table 4. Multi-view consistency versus artefact rate at different voting thresholds. Results measured on 7-object benchmark with 15 views per scene.

Voting Threshold	Consistency (%) ↑	Ghost Index (%) ↓
$N/2.0$	82.41	22.48
$N/1.8$	87.69	37.42
$N/1.5$	**93.72**	**46.27**
$N/1.2$	100.0	53.19
$N/1.0$	100.0	67.23

4.4.2. Overall semantic quality

To assess overall semantic understanding, we computed the mIoU between the projected semantic masks and manually annotated foreground masks. Our framework achieves a 2D segmentation mIoU of 0.87 averaged across all views and objects, and 3D projection consistency reaches 0.93. These results confirm that our multi-view fusion approach effectively bridges 2D perception and 3D geometric reconstruction, providing a reliable semantic layer for robotic manipulation.

4.5. Ablation study on point cloud cleaning

To validate the effectiveness of each component within our cleaning pipeline, we conducted a rigorous ablation study on four representative objects from the L1-Basic and L2-Complex categories (Blue Box, Yellow Cube, Toy Hammer, Scissors). The raw point cloud generated by 3DGS typically contains floaters and ghosting artefacts that compromise geometric accuracy. To qualitatively illustrate this issue and the effectiveness of our cleaning pipeline, we present representative results in Figure 3. Raw 3DGS reconstructions contain floating artefacts and surface fuzziness, which can create unreliable geometry for collision checking and motion planning. After applying the proposed multi-stage filtering strategy, the refined point clouds show clearer object boundaries and improved structural consistency. This qualitative comparison shows why geometric cleaning is required before converting 3DGS reconstructions into planning-ready digital twins.

Figure 3. Qualitative results of the point cloud cleaning pipeline. Top: Raw 3DGS point clouds exhibit floaters and surface fuzziness, which impede precise collision checking. Bottom: Refined geometries after applying our multi-stage filtering, including heuristic filtering and DBSCAN. The process effectively removes artefacts and sharpens boundaries, yielding planning-ready digital twins for manipulation tasks.

We compared four configurations: the original 3DGS output serving as baseline; denoising only, applying attribute-based filtering (opacity and colour thresholds); clustering only, applying DBSCAN spatial clustering (eps = 0.02, min_samples = 10); and our full method combining both components sequentially. Results are presented in Table 5.

Table 5. Ablation study on geometric fidelity of the cleaning pipeline. Results averaged across four test objects. Ground truth obtained via manual cleaning in CloudCompare. Chamfer Distance computed at 1mm resolution.

Method	Chamfer Distance ↓	Precision ↑	F1-Score ↑
Original	0.0052	0.8846	0.9369
Denoising Only	0.0055	0.8838	0.9369
Clustering Only	0.0043	0.8958	0.9429
Denoising + Clustering	**0.0020**	**0.9977**	**0.9989**

The results reveal the contribution of each component. Applying clustering alone provides noticeable improvement over baseline, reducing Chamfer distance from 0.0052 to 0.0043 and increasing F1-Score from 0.9369 to 0.9429, indicating effectiveness in removing spatially isolated noise. Interestingly, applying de-noising in isolation yields no improvement (Chamfer distance slightly increased to 0.0055), suggesting that attribute-based filtering alone cannot handle complex artefacts where floaters remain spatially connected to the main body. However, the full method achieves substantial improvements, reducing Chamfer distance to 0.0020 and elevating F1-Score to 0.9989. This demonstrates a synergistic effect: attribute-based filtering first removes low-confidence primitives, while clustering then removes spatially isolated components that remain after filtering. This two-stage process is essential for producing high-fidelity point clouds required for reliable manipulation.

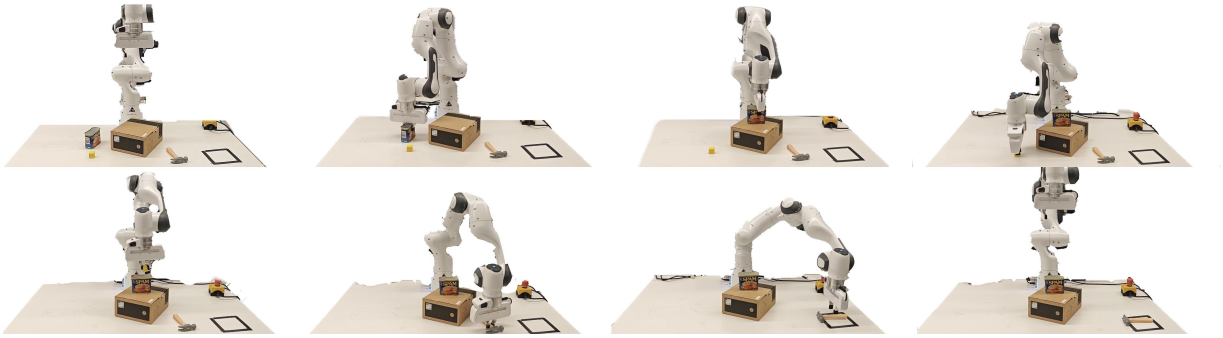
4.6. Real-world robotic validation

We finally validate this framework on a Franka Emika arm to demonstrate its ability to enable successful real-world manipulation. We conducted 10 independent trials of the long-horizon rearrangement task, with the complete execution sequence visualised in Figure 4.

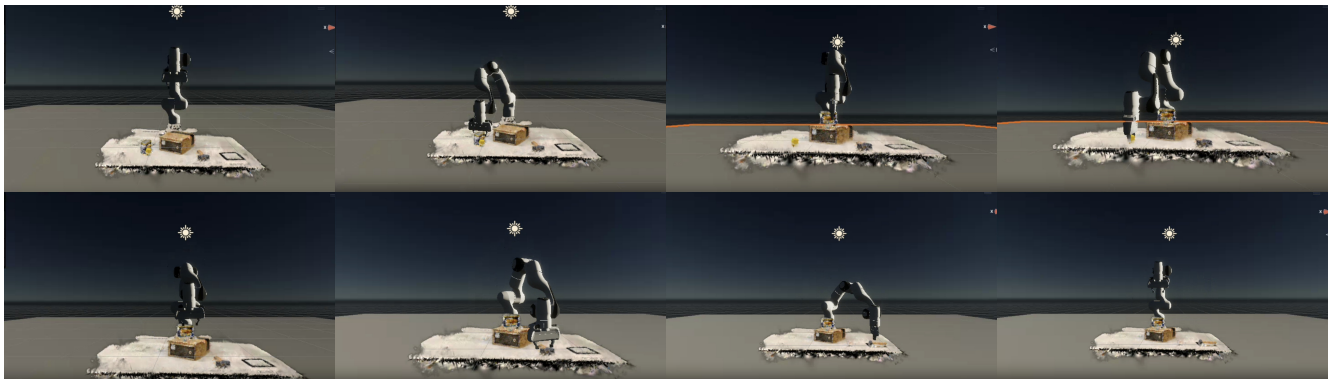
Success criteria: To ensure rigorous evaluation, we define a trial as successful only if it meets three conditions: (1) the robot successfully detects and grasps the correct target object; (2) the object is transported and placed stably within the designated goal region; and (3) the entire trajectory is collision-free with respect to both static obstacles and the environment.

Results analysis: Under these strict criteria, our framework achieved a 100% success rate in simulation validation and a 90% success rate in real-world execution (9/10 trials). The solitary failure occurred during the grasping attempt of the 2.5 cm Yellow Cube. Due to the object's diminutive scale, a minor gripper alignment error resulted in a missed grasp. This failure case highlights the high-precision challenges inherent in manipulating small-scale objects that approach the resolution limits of the 3DGS-based geometric reconstruction and gripper finger geometry.

In terms of placement accuracy, qualitative assessment confirmed that all manipulated objects were correctly deposited strictly within the designated regions. While exact metric error was not instrumented, this consistent alignment demonstrates that the system effectively satisfied the spatial tolerances required for the rearrangement task. Importantly, zero collisions were observed during any trial, validating the high fidelity of our collision geometry generation. These results demonstrate that 3DGS-based digital twins, combined with semantic and geometric consistency, provide a reliable foundation for complex manipulation in unstructured environments.



(a)



(b)

Figure 4. Execution sequence of the multi-step rearrangement task in **(a)** the real world and **(b)** the digital twin. The robot grasps the Blue Box and places it on the cardboard box, then grasps the Yellow Cube and stacks it on the Blue Box, and finally grasps the Toy Hammer and places it in the target area. This demonstrates the framework's capability for complex, zero-shot manipulation with proactive planning validated in simulation.

5. Limitations

Despite the demonstrated effectiveness of our framework, several limitations warrant acknowledgment to set realistic expectations for deployment and guide future improvements.

Static-scene assumption: Our current system performs a one-time reconstruction and assumes objects remain stationary during the planning and execution phases (typically 5–15 seconds in our experiments). This design choice prioritises reconstruction efficiency and geometric stability, which are key requirements for the proposed rapid scan-and-plan workflow. However, it limits applicability to dynamic scenarios involving deformable objects, moving humans, or objects that shift during task execution. The framework is specifically designed for tabletop rearrangement tasks where the environment remains stable during planning, a common assumption in structured manipulation settings. Extending to dynamic scenes would require integrating continuous reconstruction methods or dynamic 3DGS variants, as discussed in Section 6.

Limited view-count validation: Our experiments focus on sparse-view reconstruction using 10–20 input views, which represents the optimal trade-off between reconstruction quality and data acquisition time for desktop-scale manipulation (approximately 2–3 minutes of robotic scanning). While this range is sufficient for our target scenarios, we have not systematically validated performance with significantly higher view counts (e.g., 30–50 views) or explored the potential quality improvements from denser sampling. The scaling behaviour beyond 20 views, including potential diminishing returns in reconstruction fidelity versus increased computational cost, remains an open empirical question.

Scene scale constraints: The framework is optimised for desktop-scale manipulation scenarios with workspace dimensions of approximately 0.3–1.0 m depth and objects ranging from 5–30 cm in size. Our DBSCAN clustering parameters (eps = 0.02 m, min_samples = 10) and semantic fusion thresholds are calibrated for this scale. Deployment in substantially different environments—such as warehouse-scale logistics (multi-meter workspaces), micro-manipulation (< 1 cm objects), or outdoor unstructured settings—would require recalibration of spatial thresholds and potentially architectural modifications to handle the increased scene complexity and point cloud density.

Object geometry constraints: The current pipeline is optimised for rigid, opaque objects with diffuse or mildly specular surfaces, as evidenced by our test objects (boxes, tools, textured bottles). Highly challenging cases such as transparent objects (glass containers), highly reflective surfaces (polished metal), thin wire-like structures (< 5 mm diameter), or materials with complex subsurface scattering may produce incomplete 3DGS reconstructions or inaccurate collision geometry. These failure modes stem from fundamental limitations in multi-view RGB-based reconstruction rather than our specific processing pipeline, but they constrain the generality of the approach.

6. Conclusion and future direction

This paper presented a holistic, closed-loop framework for rapidly creating high-fidelity, interactive digital twins for robotic manipulation from sparse RGB views. The approach combines 3DGS [6] for fast photorealistic reconstruction, Grounded SAM [8] for zero-shot semantic segmentation, and a filtering pipeline to generate clean, planning-ready collision geometry. The system completes reconstruction in

under 4 minutes on average (229 s) while achieving high visual fidelity (37.03 dB PSNR and 0.9821 SSIM), representing a 5-fold speed-up over the NeRF-based baseline.

Real-world experiments on a long-horizon rearrangement task demonstrate the practical utility of the framework. Motion plans generated and validated within the digital twin achieved a 90% success rate when executed on a Franka Emika robot, with all successful placements falling within the predefined task regions. These results provide evidence that the framework effectively addresses the sim-to-real gap, enabling reliable robot operation in unstructured environments.

The multi-stage de-noising and meshing pipeline proved essential for converting the unstructured 3DGS output into planner-compatible geometries. The ablation study confirms that both heuristic filtering and cluster-based de-noising contribute synergistically, achieving a near-perfect F1-score of 0.9989 against manually cleaned ground truth models. This Gaussian-to-mesh conversion represents a critical bridge between modern neural rendering and traditional motion planning frameworks.

Several directions warrant future investigation. First, the current framework assumes static scenes and performs a one-time reconstruction. Integrating dynamic 3DGS variants [25] or implementing continuous update mechanisms would enable the digital twin to remain consistent with evolving environments. Second, the system currently models only geometry and appearance. Incorporating methods for online physical property estimation [26] would enable more sophisticated manipulation strategies involving contact-rich interactions. Third, while this work focuses on motion planning given predefined grasps, integrating robust grasp planning modules [18] that operate directly on Gaussian representations would further automate the pipeline.

A particularly promising direction involves leveraging the digital twin as an enabler for learned policies. The framework could serve dual purposes: as a safety validation platform where policies are tested through thousands of simulated iterations before deployment, and as a data generation engine that autonomously creates diverse training datasets without physical resource consumption. This capability could significantly accelerate the development of vision-language-action models and reinforcement learning approaches for manipulation.

The work demonstrates that the synergy between 3DGS efficiency, foundation model capabilities, and robust geometric processing provides a practical paradigm for robotic manipulation in unstructured environments. By unifying perception, reconstruction, and planning into a closed-loop system, the framework represents a step toward autonomous robots that can rapidly adapt to novel surroundings with both speed and reliability.

Data availability statement

To support reproducibility, all code, datasets, and documentation are available at: https://github.com/535A59/3DGS-Digital-Twin.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used ChatGPT only to improve language and readability in limited sections of the manuscript. This tool was not used for scientific content generation,

data analysis, result interpretation, figure generation, or idea development. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of this manuscript.

Acknowledgments

This work was partially supported by the Advanced Research and Invention Agency [grant number SMRB-SE01-P06] and NVIDIA Academic Grant Program.

Authors' contribution

Ziyang Sun: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, visualisation and writing—original draft; Lingfan Bao: methodology, validation, and writing—review and editing; Tianhu Peng: writing—review and editing; Jingcheng Sun: writing—review and editing. Chengxu Zhou: conceptualisation, resources, supervision, project administration, funding acquisition and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no competing interests.

References

[1] Li J, Yang SX. Digital twins to embodied artificial intelligence: review and perspective. *Intell. Robot.* 2025, 5(1):202–227.

[2] Zhou L, Wu G, Zuo Y, Chen X, Hu H. A comprehensive review of vision-based 3D reconstruction methods. *Sensors* 2024, 24(7):2314.

[3] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, *et al.* NeRF: representing scenes as neural radiance fields for view synthesis. *arXiv* 2020, arXiv:2003.08934.

[4] Barron JT, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R, *et al.* Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. *arXiv* 2021, arXiv:2103.13415.

[5] Lv C, Lin W, Zhao B. Voxel structure-based mesh reconstruction from a 3D point cloud. *IEEE Trans. Multimedia* 2022, 24:1815–1829.

[6] Kerbl B, Kopanas G, Leimkühler T, Drettakis G. 3D Gaussian Splatting for real-time radiance field rendering. *arXiv* 2023, arXiv:2308.04079.

[7] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, *et al.* Segment anything. *arXiv* 2023, arXiv:2304.02643.

[8] Ren T, Liu S, Zeng A, Lin J, Li K, *et al.* Grounded SAM: assembling open-world models for diverse visual tasks. *arXiv* 2024, arXiv:2401.14159.

[9] Chen T, Shorinwa O, Bruno J, Swann A, Yu J, *et al.* Splat-Nav: safe real-time robot navigation in Gaussian Splatting maps. *arXiv* 2025, arXiv:2403.02751.

[10] Shorinwa O, Tucker J, Smith A, Swann A, Chen T, *et al.* Splat-MOVER: multi-stage, open-vocabulary robotic manipulation via editable Gaussian Splatting. In *The 8th Conference on Robot Learning (CoRL 2024)*, Munich, Germany, November 6–9, 2024.

[11] Li X, Li J, Zhang Z, Zhang R, Jia F, *et al.* RoboGSim: a Real2Sim2Real robotic Gaussian Splatting simulator. *arXiv* 2025, arXiv:2411.11839.

[12] Ravi N, Gabeur V, Hu Y, Hu R, Ryali C, *et al.* SAM 2: segment anything in images and videos. *arXiv* 2024, arXiv:2408.00714.

[13] Coleman D, Sucan I, Chitta S, Correll N. Reducing the barrier to entry of complex robotic software: a MoveIt! Case study. *arXiv* 2014, arXiv:1404.3785.

[14] Curless B, Levoy M. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New Orleans, USA, August 4–9, 1996, pp. 303–312.

[15] Oleynikova H, Taylor Z, Fehr M, Siegwart R, Nieto J. Voxblox: incremental 3D Euclidean signed distance fields for on-board MAV planning. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, September 24–28, 2017, pp. 1366–1373.

[16] Müller T, Evans A, Schied C, Keller A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graphics* 2022, 41(4):1–15.

[17] Cen J, Fang J, Yang C, Xie L, Zhang X, *et al.* Segment any 3D Gaussians. *arXiv* 2025, arXiv:2312.00860.

[18] Ji M, Qiu R, Zou X, Wang X. GraspSplats: efficient manipulation with 3D feature splatting. *arXiv* 2024, arXiv:2409.02084.

[19] Fan Z, Wen K, Cong W, Wang K, Zhang J, *et al.* InstantSplat: sparse-view Gaussian Splatting in seconds. *arXiv* 2025, arXiv:2403.20309.

[20] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Cagliari, Italy, July 4–6, 2006, pp. 61–70.

[21] Guédon A, Lepetit V. SuGaR: surface-aligned Gaussian Splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. *arXiv* 2023, arXiv:2311.12775.

[22] Pranckevicius A. UnityGaussianSplatting. 2024. Available: https://github.com/aras-p/UnityGaussianSplatting (accessed on 28 April 2025).

[23] RobotecAI. ROS2 for unity. 2024. Available: https://github.com/RobotecAI/ros2-for-unity (accessed on 28 April 2025).

[24] Leroy V, Cabon Y, Revaud J. Grounding image matching in 3D with MASt3R. *arXiv* 2024, arXiv:2406.09756.

[25] Zhou X, Lin Z, Shan X, Wang Y, Sun D, *et al.* DrivingGaussian: composite Gaussian Splatting for surrounding dynamic autonomous driving scenes. *arXiv* 2023, arXiv: 2312.07920.

[26] Cherian A, Corcodel R, Jain S, Romeres D. LLMPhy: complex physical reasoning using large language models and world models. *arXiv* 2024, arXiv: 2411.08027.