RealTwin: Concept Graph Representation and Grounding Framework for Reality-Preserving Digital Twin Reconstruction

Zisu Li
The Hong Kong University of Science and Technology
Hong Kong SAR, Hong Kong, China
zlihe@connect.ust.hk

Ruohao Li
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
rli777@connect.hkust-gz.edu.cn

Jiawei Li
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
jli526@connect.hkust-gz.edu.cn

Chao Liu
Mechanical Engineering
The University of British Columbia
Vancouver, British Columbia, Canada
chao.liu@ubc.ca

Junyi Zhu
EECS
University of Michigan
Ann Arbor, Michigan, USA
zhujunyi@umich.edu

Daniela Rus
Distributed Robotics Lab
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
rus@csail.mit.edu

Chen Liang*
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
lliangchenc@163.com

Mingming Fan*
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
The Hong Kong University of Science and Technology
Hong Kong, China
mingmingfan@ust.hk

Abstract

Reconstructing realistic digital twins has become crucial as advances in mixed reality, metaverse, and robotics demand more accurate simulations for the physical world. Despite technical progress, building high-fidelity digital twins from a systematic and human-centered perspective remains underexplored. Drawing from the human processing model, we decompose human-centric reality into perception, motion, and cognition, and define a reality-preserving digital twin (RPDT) as a reconstruction integrating these dimensions. We present RealTwin, an attribute-graph-based representation and inference framework for RPDT. Leveraging the grounding capabilities of Multimodal Large Language Models (MLLMs), RealTwin chains AI tools to construct attribute graphs that faithfully encode real-world properties. We validate RealTwin through both technical evaluation, showing promising success in graph parsing and attribute inference, and a user study, assessing its applicability across diverse user groups. Enlightened by RealTwin, we discuss critical issues, including ecology, interaction space, and real-world adoption, for future end-to-end, fine-grained, and scalable digital twin reconstruction.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**.

Keywords

Digital Twin, Digital Content Creation, 3D Reconstruction

ACM Reference Format:
Zisu Li, Ruohao Li, Jiawei Li, Chao Liu, Junyi Zhu, Daniela Rus, Chen Liang, and Mingming Fan. 2026. RealTwin: Concept Graph Representation and Grounding Framework for Reality-Preserving Digital Twin Reconstruction. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3772318.3790590

1 Introduction

Realistic digital twins have become increasingly important across a wide range of applications, serving as the simulation of the physical world in many tasks such as robot training [46, 64], metaverse content creation [44], and VR/AR interactions [27, 42, 69]. Constructing high-fidelity digital twins that faithfully reflect the object's real-world properties, geometry, appearance, physical properties, and interactivity, has long been a challenging problem that researchers in human-computer interaction, computer vision, computer graphics, and robotics are collectively working on.

Existing research has covered the reconstruction of different properties that make the digital twin "more realistic", including precise 3D geometry [6, 86], photorealistic texture [16, 98], deformable material [46, 87], and mechanical structure [37, 61]. While these techniques have achieved superior performance in domain-specific tasks, they often rely on large-scale, high-quality, and task-specific

*Corresponding authors.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3790590

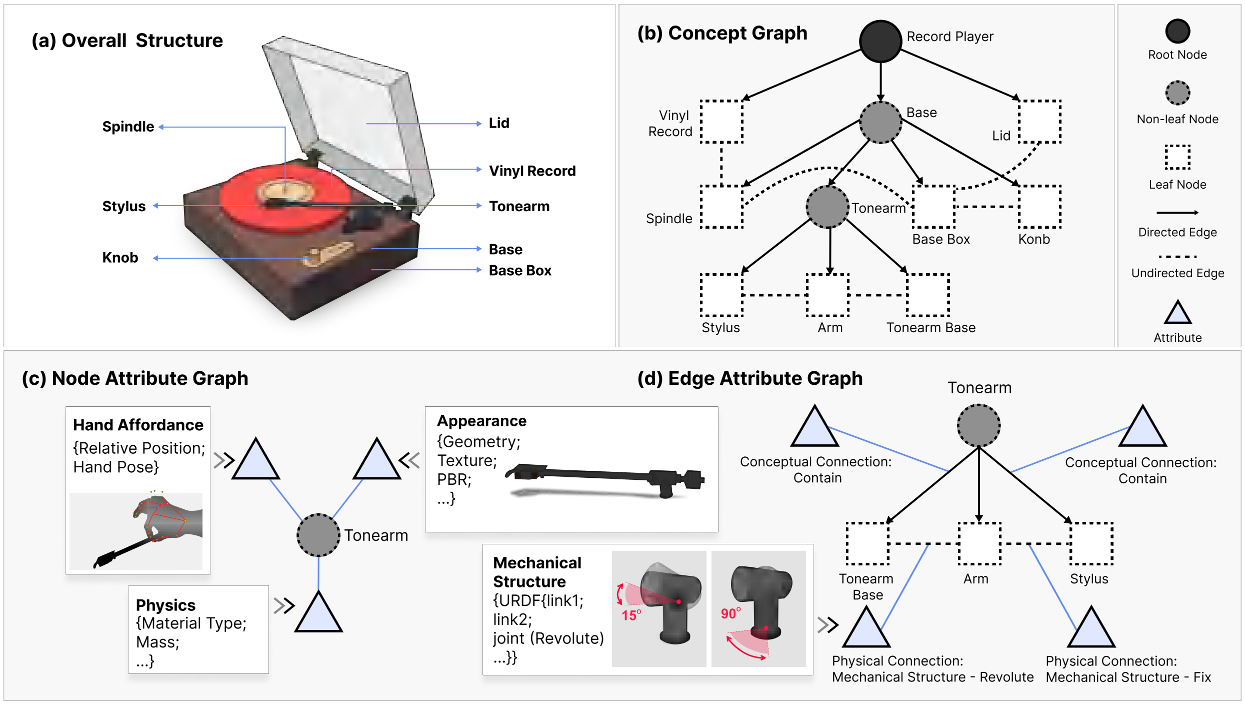


Figure 1: A concept graph representing a physical object's component-level real-world properties to guide its digital-twin reconstruction. The record player in (a) is parsed into the concept graph shown in (b). (c) shows the tonearm's node-attribute graph, including hand affordance, appearance, and physics. (d) depicts the tonearm's edge types and their attributes: the conceptual connections, shown as directed edges that link each parent vertex to its child vertices, and physical connections, shown as undirected edges that link only leaf vertices with mechanical structure attributes.

datasets in the target domain to train a task-specific model. Such objectives become even harder when it comes to the physical domain, where high-fidelity 3D data with detailed physical and mechanical annotations is costly to acquire and challenging to scale up. Moreover, although existing work has extensively focused on specific technical points around "realism" [46, 87], few have delved into the true meaning of "realistic digital twin reconstruction" from a systemic and human-centered perspective.

Recently, Multimodal Large Language Models (MLLMs) have offered promising new advances that could make a step forward in addressing these challenges. MLLMs have shown strong reasoning capabilities in open-world tasks involving complex and intertwined physical and semantic information, such as scene understanding [19, 45], robot task planning [56], and interaction grounding [48]. In these tasks, the MLLM, equipped with an extensive physical knowledge base, can perform cognition, perception, and reasoning on complex information like humans and carry out task grounding. For instance, except for the appearance and the material types, MLLM could identify implicit and unseen interactions such as affordances (e.g., knowing which parts of an object can be grasped, pressed, or rotated).

In this paper, we present a representation and grounding framework, RealTwin, to reconstruct digital twins with reality-preserving properties from a human-centric perspective, leveraging MLLM's strong capability in real-world physical reasoning. Drawing inspiration from the human processor model [11], we conceptualize human-centric reality in three dimensions: (1) perception reality, (2) motion or interaction reality, and (3) cognition reality. Building upon these dimensions, we first define the concept of a reality-preserving digital twin (RPDT) as an entity that integrates these characteristics into the reconstruction of digital twins. Then we introduce a concept graph representation approach that maintains component-level details of real-world properties, providing both structural clarity and semantic richness. Such a representation aligns well with MLLM-based semantic inference and benefits from its flexibility and scalability. Furthermore, we propose an MLLM-driven grounding framework that enables the automatic reconstruction of RPDT from scratch, encompassing both (1) the grounding of graph structures and (2) the inference of attributes within complex data representations. To enhance the applicability of our framework for end users, we provide interfaces that enable them to edit intermediate results, allowing users to better control the reconstruction of RPDTs.

To evaluate the feasibility of leveraging MLLM to construct the object-centric graph and the efficacy of automatic framework integrated with AI tool chaining, we conduct an evaluation on 50 diverse objects and achieve a high F-1 score for concept graph construction (98.3% for vertices and 92.2% for edges), material recognition

(94.4%), mechanical structure recognition (97.1%), and hand affordance inference (92.5%). The results demonstrate that, under our proposed digital twin representation format, MLLMs exhibit strong zero-shot grounding capabilities for real-world objects across appearance, physics, functionality, and interactivity. Moreover, these grounding capabilities can be further implemented with external AI tools chained by the MLLM.

To assess the practical applicability of our framework, we conducted a user study with participants from diverse professional backgrounds. Through the provided interfaces, participants reconstructed digital twins and reported that RealTwin was intuitive to use, expressive in representing concepts, and engaging throughout the process. They highlighted the richness of the RPDT concept, noted the framework's future extensions for improving interaction fidelity and structural complexity, and suggested design implications for future development in domains such as education, digital art, fashion design, and sports.

The contributions of this paper are threefold: 1) We introduce the concept of the Reality-Preserving Digital Twin (RPDT), derived from a human-centric perspective, together with a scalable concept graph representation method; 2) We propose an MLLM-driven framework with AI tool chaining for grounding the structure and inferring the attributes of RPDT; 3) We present both a technical evaluation of MLLM's zero-shot grounding capability for RPDT and a user study to demonstrate the practical applicability and future opportunities of RealTwin.

2 Related Work

2.1 Digital Twin Reconstruction for Real-World Objects

Digital twins simulating the physical-world objects have been increasingly prominent across various fields, such as motion planning and manipulation in robotics [46], content creation in metaverse [2, 44], and interaction generation or prediction in virtual reality [1, 25, 42, 49, 71]. This growth is driving the demand for advanced 3D reconstruction technologies for reconstructing more realistic digital twins in an automatic way for physical-world objects. Prior work attempted to duplicate the geometry and the appearance of an object for its digital twin [6, 86]. For example, *SF3D* proposed a method for rapid and high-quality textured object mesh reconstruction [6]. However, these static 3D representations of physical objects cannot be directly applied to robot simulation tasks or virtual interactive scenarios, which require the digital copies of objects to be functional, or in other words, to be able to respond to external stimuli in a way that aligns with the physical world. To this end, recent work attempted to predict and simulate the physical properties of objects through visual priors. For example, *PhysGaussian* seamlessly integrates physically grounded Newtonian dynamics within 3D Gaussians to achieve high-quality object motion synthesis [87]. *PhysDreamer* endows static 3D objects with interactive dynamics by leveraging the object dynamics priors learned by video generation models [96]. *VR-GS* enables real-time execution with realistic dynamic responses on virtual objects by developing a physical dynamics-aware interactive Gaussian Splatting in a VR setting [49]. *PhysTwin* proposed a novel framework that uses sparse videos of dynamic objects under interaction to produce a photo- and physically realistic, real-time interactive virtual replica [46]. These learned models depend on pre-labeled and specified datasets, requiring precise specification of boundary conditions or material properties of the object to be simulated as a prerequisite in addition to the visual input.

Additionally, visual priors are typically limited to reasoning about objects made from uniform materials. For objects consisting of multiple components with varying materials (e.g., a bottle with a plastic body and a silicone cap, even if they share the same color), or for objects featuring complex internal mechanical structures (e.g., a sippy cup), visual data for the whole object's appearance alone cannot fully capture the implicit material and structural details. However, this information can be easily parsed by MLLMs, which are equipped with extensive knowledge of the production and construction of physical objects [26, 56]. Recent work has explored using MLLMs to infer inner-object structure, including articulation generation (e.g., PhysX-Anything [10], URDF-Anything [59], ArticulateAnything [54]), part-level physical reasoning (e.g., PhysX-3D [9], MeshLLM [31], LLaMA-Mesh [83], OmniObject3D [85]), and affordance description (e.g., PhysX-Anything [10], PhysX-3D [9]). These approaches demonstrate the potential of multimodal models for predicting geometry and physical properties, but they remain primarily generative and dataset-driven, and typically do not produce a systematic, human-interpretable representation that supports downstream editing or interactive refinement. RealTwin provides a systematic digital-twin reconstruction pipeline that goes beyond specific property prediction. It outputs a unified attribute graph to guide reconstruction, and supports automatic articulation extraction, structural parsing, affordance generation, and rig synthesis. Moreover, RealTwin operates in a zero-shot manner while also supporting an optional human-in-the-loop interactive authoring workflow, enabling non-expert users to create and refine digital twins without requiring domain-specific prior knowledge.

2.2 Realism in Digital Representations from Human-Centered Perspectives

Realism is a crucial measure for digital representations of the physical world. While advances in techniques have enabled highly detailed 3D geometries [6, 86], photorealistic rendering [98], and accurate physical modeling [46, 87], these forms of technical realism do not necessarily translate into what users perceive as real [28, 29]. For instance, even with photorealistic shading, digital humans with rigid or unnatural expressions often evoke discomfort, a phenomenon widely referred to as the uncanny valley [67]. This gap highlights that realism is not only a matter of technical fidelity but fundamentally a matter of human judgment.

Prior work has emphasized that realism in digital environments is constructed through the interpretive processes of human perception and cognition [93]. On the perceptual side, the human visual system is highly sensitive to cues such as lighting consistency, motion continuity, and material detail [28, 33, 93]. On the cognitive side, realism is further shaped by factors including cultural background, prior experience, and situational context [5]. For instance, when seeing a cup, people immediately associates it with holding liquid; similarly, when a cup rendered with glass-like material appears to fall, people naturally anticipate it might break. Thus, establishing a human-centered concept of realism in digital twins can significantly blur the boundary between the virtual and the real.

This perspective is particularly important in application domains such as virtual reality [20, 33, 50, 70], education [4, 82, 97], and human–robot interaction [32]. When digital representations fail to align with human perceptual expectations, task performance, sense of presence, and user trust are easily compromised [32, 79, 80]. Conversely, understanding the perceptual attributes that most strongly contribute to realism—such as the motions [32], the behaviors or interactions [38, 43, 78], or the tactile plausibility [81, 94]—enables systems to improve user experience directly.

Inspired by the Human Processor Model [11], which frames perception, action, and cognition as integral components of human experience, we systematically characterize the human-centered attributes that influence realism in digital representations. Building on this framework, we decompose human-centered realism in digital twins into three corresponding dimensions—perceptual reality, motion or interaction reality, and cognitive reality—and integrate them into our proposed concept of the reality-preserving digital twin.

2.3 MLLM Grounding for the Physical World

Multi-modal Large Language Models (MLLMs) have emerged as powerful tools for understanding and interpreting the physical world through comprehensive multimodal reasoning [15, 22, 56, 99]. By leveraging large-scale visual and textual corpora, MLLMs can go beyond simple tasks such as object recognition, scene description, and contextual queries [21, 30, 52]. They are also capable of extracting attributes of objects from an image, enabling deeper physical world grounding. The SayCan series, for instance, has demonstrated how high-level semantic abstractions that extend beyond raw pixels can be effectively linked to real-world environments. [7, 14, 23, 41, 91].

Furthermore, MLLMs offer a zero-shot and scalable framework for world understanding, and are easy to generalize to new tasks and environments without requiring task-specific data or retraining [77, 88]. This generalization is often enabled by the model's ability to incorporate rich object-centric contextual information (e.g., object co-occurrence patterns [63], affordance priors [74, 89], and common-sense physical knowledge [3]. By embedding the knowledge into object representations, MLLMs can reason more effectively about object identity, functionality, and interaction dynamics [39]. This makes them particularly well-suited for tasks that require structured and grounded object understanding. Thus, in this work, we apply MLLMs to generate and construct a computational and scalable attribute graph for digital twin reconstruction.

In addition to MLLMs, knowledge graph-based representations have become increasingly important for 3D modeling and physical-world tasks, such as virtual object retrieval [47, 48], human-object interaction understanding [23, 58], scene understanding [19, 26, 36, 45, 56], and robot task planning [56], providing an expressive and flexible way to represent complex information. This graph representation method has increasingly gained attention to enhance MLLMs with structured, interpretable outputs. These graphs make outputs from MLLMs computationally tractable, easily editable, and intuitive to visualize [56], which is essential for downstream tasks [12, 56]. Inspired by these methods of MLLM-driven graph representation, we apply MLLMs to parse implicit contextual object information to construct the object-centric attribute graph framework, making reality-preserving digital twins informative and scalable for downstream tasks.

3 Representing and Grounding Reality-Preserving Digital Twins (RPDTs) with Concept Graph

In this section, we first define the concept of Reality-Preserving Digital Twin (RPDT) and the structural representation using an object-centric concept graph for RPDT. Then, we present an MLLM-based grounding framework to automatically construct RPDT.

3.1 Definition and Implications of RPDT

The concept of reality-preserving centers around the relation, interaction, and ecology between the object and its surrounding environments. State-of-the-art game engines (e.g., UE 5) can achieve both fine renderings in appearance (e.g., with industrial PBR production) and certain degrees of interactivity beyond static meshes (e.g., adding physical properties to the target object), which push forward the boundary of "realism" in the virtual space. However, they often rely on customizations in engineering for certain effects without a unified standard. From a human-centric perspective, the representation of a digital object that makes itself "real" should conceptually align well with the human's understanding, perception, and expectation of the real copy. Consequently, a key question to be unveiled is how we can define the boundaries of "realism" from the human end.

Inspired by the Human Processor Model [11], which frames perception, action, and cognition as the core of human experience, we decompose human-centered realism into three corresponding aspects: perceptual, motion, and cognitive systems, which shape how reality is experienced. Reflections on these layers are: **(1) Perception reality** means how closely the visual appearance of a digital object mimics its real-world counterpart. High-fidelity geometry, texture, and material attributes can significantly impact the effect of photorealistic rendering. Unfortunately, in the vast collection of web images and 3D assets, fine-grained appearance attributes (e.g., materials of an object's different segments) are not prioritized and are often missing in the majority. **(2) Motion, or interaction reality** stands for creating a faithful physics simulation where the digital object behaves as its real-world equivalent would, especially in how it responds to environments and human operations. This involves the representation and simulation of an object's interactive features, such as mechanical structures, soft materials, and articulations, which are key to making a digital object interactable, though usually too complex to be explicitly annotated. **(3) Cognition reality** involves the higher-level cognitive processes where users mentally model and understand the appearance, interactivity, and functionality based on their observations of and interactions with a digital object.

Consequently, we define the Reality-Preserving Digital Twin (RPDT) as a digital counterpart that is cognitively, visually, and interactively consistent with reality. Given a real-world object, we expect its RPDT to be constructed and maintained structurally, with dynamic and scalable attributes and details for appearance, physics, and interactivity.

3.2 Concept Graph Representation for RPDT

To holistically model the above-mentioned multidimensional realities, we propose a graph structure that hierarchically integrates appearance attributes (e.g., material, geometric segmentation) as the perception reality [72]; physical properties (e.g., mechanical structure, kinematic, articulation, mass distribution) correspond to the cognitive reality for reasoning about object dynamics [51]; and interaction semantics (hand affordances, causal relationships) lead to motion or interaction reality (physical interactions, manipulation) [34, 68], as is shown in Fig. 1. This graph acts as a unified knowledge backbone that dynamically aligns human expectations with computational representations. In addition, it ensures scalability through modular updates of nodes (object components) and edges (inter-component dependencies or environmental interactions) while preserving cross-modal consistency across perception, motion, and cognition layers.

3.2.1 *Graph Elements.*

The concept graph of RPDT is built upon conceptual subdivisions where a particular part of an object is obliged to be distinguished from another. As is shown in Fig. 1(b), starting from the root node that represents the object itself, the graph illustrates a series of physical subdivisions that gradually break the object into atomic components with indivisible physics and semantics. In the concept graph $G_{RPDT} = (V, E)$, a node $v \in V$ represents a component of the object at a specific subdivision layer (the object itself is also a component of its own), while an edge $e \in E$ linking two nodes represents the physical or conceptual relation between them.

Vertices can be categorized into three types - root, non-leaf, and leaf vertices, as is shown in Fig. 1(b). The root vertex refers to the object itself, which maintains the global information of the object. Non-leaf vertices represent intermediate components that can be further subdivided into smaller parts, while leaf vertices represent the atomic components that cannot be further subdivided and have indivisible physics and semantics. A vertex, regardless of its type, can be represented as $v = (p^*, C, A_v, R)$, where p^* refers to a pointer to its parent vertice; $C = \{c_1^*, c_2^* \cdots\}$ refers to a list of pointers to its child vertices (subdivisions); A_v refers to an attributed dictionary maintained at this vertice; R refers to a list of relations that it has with other vertices.

There are two types of edge: 1) a directed edge that indicates a parent-child relationship, which means one component is the direct subdivision of the other, and 2) an undirected edge that indicates a physical relation among two components. The former type is simply a pointer that indicates a subdivision relationship, while an edge $e = (v1, v2, A_e)$ of the latter type contains a dictionary of attributes A_e that characterize the physical properties of the connection $(v1, v2)$.

By expanding the attribute space A_v, A_e in vertices and edges, we can easily extend realistic appearance, physics, and interactive features associated to particular object components. Below, we illustrate some key attributes for reality-preserving reconstruction.

3.2.2 *Appearance, Physics, and Mechanical Structures of RPDT.*

Each component's appearance attributes mainly include geometry (e.g., mesh), textures, and physically-based rendering (PBR) material properties. These attributes are mostly compatible with the representations in advanced photorealistic rendering engines such as Blender and Unreal Engine 5, so that cutting-edge rendering techniques, such as LOD optimization for watertight mesh and texture, can be implemented based on our representations.

Physical attributes are parameters that illustrate the object's physical properties in materials, including material type (e.g., metal, plastic, rubber, clay, etc.), roughness, mass distribution, and material-specified mechanics parameters (e.g., Young's modulus for deformable bodies, yield stress for plasticity, Rayleigh damping coefficients, etc.). These attributes can better help to simulate the physical behavior and interactions of objects within the virtual environment, enabling accurate modeling of their deformation and dynamic response to external forces.

For mechanical structures, the attribute representation in RPDT is designed to align with widely adopted robotic modeling standards such as URDF (Unified Robot Description Format), ensuring seamless compatibility with existing robotics frameworks like ROS (Robot Operating System). Each mechanical component's attributes include joint definitions (e.g., revolute, prismatic, fixed), kinematic chains, and dynamic constraints—all structured in a format directly translatable to URDF specifications. For instance, a revolute joint between two components would encode its axis of rotation, angular limits, and torque parameters within the edge attribute dictionary. Moreover, complex attributes such as rigging, where the mechanics are coupled with materials, can also be introduced to simulate advanced physical effects such as soft limb movements and anisotropic force response.

The above attributes are anchored to particular vertices (e.g., appearance attributes, physical attributes, and articulations) or edges (e.g., URDF joints).

3.2.3 *Affordance and Interactivity of RPDT.*

In addition to the above attributes to characterize an object itself, affordance and interactivity, or how objects can be manipulated and interacted with by humans, are also critical attributes in building RPDT [35, 68]. Affordance of a component can be formalized through tuple descriptors (G, E), where G represents the target gesture to trigger the affordance, which can be either a static hand pose p (e.g., represented in 21 keypoints [1] or MANO hand pose [75]) or a sequence of poses $(p_1, \cdots, p_K)$ under the component mesh's coordinates. A threshold-based method measuring whether $||p' - p||_2 < \tau$ can be applied to determine whether an arbitrary hand pose p' matches a target hand pose p in the component mesh's frame. E refers to a program (or a script) binding to the component that will be triggered upon the gesture's matching and unmatching events. For example, E may encode (1) a rotational update that opens a hinged lid when a pinch–lift gesture is detected, or (2) a state-change script that dispenses liquid when a wrap-grasp gesture anchors around a bottle's pump head. This enables users to engage with an object in a manner that reflects real-world interactions with component-level responsiveness, which can better reflect the interaction behavior with complex objects. Since the capability of E is constrained by

[1] https://github.com/google-ai-edge/mediapipe

the underlying simulation engine (e.g., fluid splashing effects are often difficult to realize in traditional game engines), for simplicity, we only consider physical anchoring behaviors in our implementation (e.g., setting a specific rigid-body part to be anchored or disanchored to particular hand segments).

3.3 Grounding RPDTs on Concept Graph

Based on the above graph representation, we develop a grounding framework that supports the automatic reconstruction of RPDT. As illustrated in Fig. 2, the workflow consists of two stages: **S1:** given an image of the target object as input, the MLLM constructs a graph representation under our prompt-engineering strategy; **S2:** and this output graph serves to inform AI tool chaining in implementing the attributes of the RPDT. We also introduce the end-user interface designed to support users in editing the intermediate results within the grounding framework.

3.3.1 Building Concept Graph with MLLM. We employ a few-shot template prompting approach [84] to decompose the object, constructing the object's concept graph. Specifically, the first author conducted a set of preliminary prompt designs and tested them with MLLM. Then, another three authors acted as quality checkers, providing feedback on the generated content. This iterative process continued through multiple rounds until all authors were satisfied with the generated content. Finally, we propose a set of four key prompt input factors, and we then concatenated each part to form a complete prompt. We introduce the rationales about how we designed the prompt below, with Fig. 3 showing its details:

(1) Task setup. This part instructs the role of MLLM (as an object decomposing assistant), its responsibilities, and its output style.

Table 1: Node and Edge Types of RPDT

Category	Type	Description
Nodes	Root node	The object itself, maintaining the global information of the object.
	Non-leaf node	Intermediate components that can be further subdivided into smaller parts.
	Leaf node	Atomic components that cannot be further subdivided and have indivisible physical and semantic properties.
Edges	Direct edge	A parent–child relationship.
	Undirected edge	A physical coupling relationship.

(2) Description of object decomposition based on RPDT. In this part, we integrate the concept of RPDT knowledge defined in Sec. 3.2 to guide the MLLM. First, we instruct the MLLM to identify the object and provide a general description. Then, we input an RPDT knowledge table (see Table 1) and provide an RPDT graph of a record player as an example (see Fig. 1) to inspire the MLLM to construct the graph. We also specify URDF as the output format for MLLM's description of mechanical structures, ensuring compatibility with subsequent AI tool processing.

(3) Output optimization. This part is designed to optimize the output generated by the MLLM through stepwise constraints and examples. We instruct the MLLM to convert the text description of the graph generated in the previous step into JSON format to facilitate reading by AI tool chaining. We also input a JSON file as an example in this part.

(4) Self-check. This part enables MLLM to check the results of its RPDT object graph construction and ensure that each component of the object has been accurately recognized and the graph elements have been successfully generated.

3.3.2 Inferring Attribute Values with AI Tool Chaining. After determining the graph structure (vertices and edges) and all the conceptual attributes in the text with MLLM inference, we employ an AI tool chaining approach to infer those attribute values that cannot be directly given by the MLLM output. We explain how we chain different AI tools with MLLM to infer several key attribute values with complex data types, including 3D mesh, mechanical structure, hand affordance, and body rig.

A1: Mesh reconstruction and subdivision. As is shown in Fig. 4(a), given one or more images of the object to reconstruct, the system first employs TRELLIS [86] to predict the entire 3D mesh of the object and assign it to the mesh value of the root vertex. Then it traverses the RPDT graph constructed from stage 1, recursively from the root vertex. At each component vertice, assuming its mesh value is known as $M = (V_M, E_M)$, the system runs a subdivision process to assign the mesh values of all its child vertices. To achieve this, the system first generates a set of orthogonal projection maps $I_1, I_2, \cdots, I_K$ with multiple views by re-imaging the parent component's mesh from uniformly distributed cameras $C_1, C_2, \cdots, C_K$. For each child component, the system acquires the child component's text description given by the MLLM from the first stage, and then employs Grounded-SAM [76], a text-guided segmentation model, on each projection map to get the segmentation masks $Mask_k$ of the target child component in different views. The system employs a slight dilation on each segmentation mask and examines each vertex's occupancy in each mask. By merging the valid mesh vertices predicted by all the masks, we can finally get the subset of mesh vertices that belong to the child component. The formulation can be represented as follows:

$$V' = \bigcup_i \{\mathbf{OrthProj}(C_i, v) \in \mathbf{dilate}(Mask_i, \epsilon), v \in V\} \tag{1}$$

where **OrthProj** refers to the orthogonal projection operation that project v to the C_i's imaging plane; **dilate** means the dilating the area of $Mask_i$ by ϵ.

A2: Mechanical structure reconstruction. We propose the mechanical structure reconstruction tool chain as is shown in Fig. 4(b). Mechanical structure is an attribute for edges that indicates the physical connection. The former stage iteratively constructs the graph with semantic annotations for every edge (e.g., the connection type). However, for physical connections, their physical attributes, including the joint position, rotation, and constraints, are not yet determined. To obtain these attributes, after reconstructing the 3D model, the system renders it into multi-view images. These rendered images are then provided to a state-of-the-art

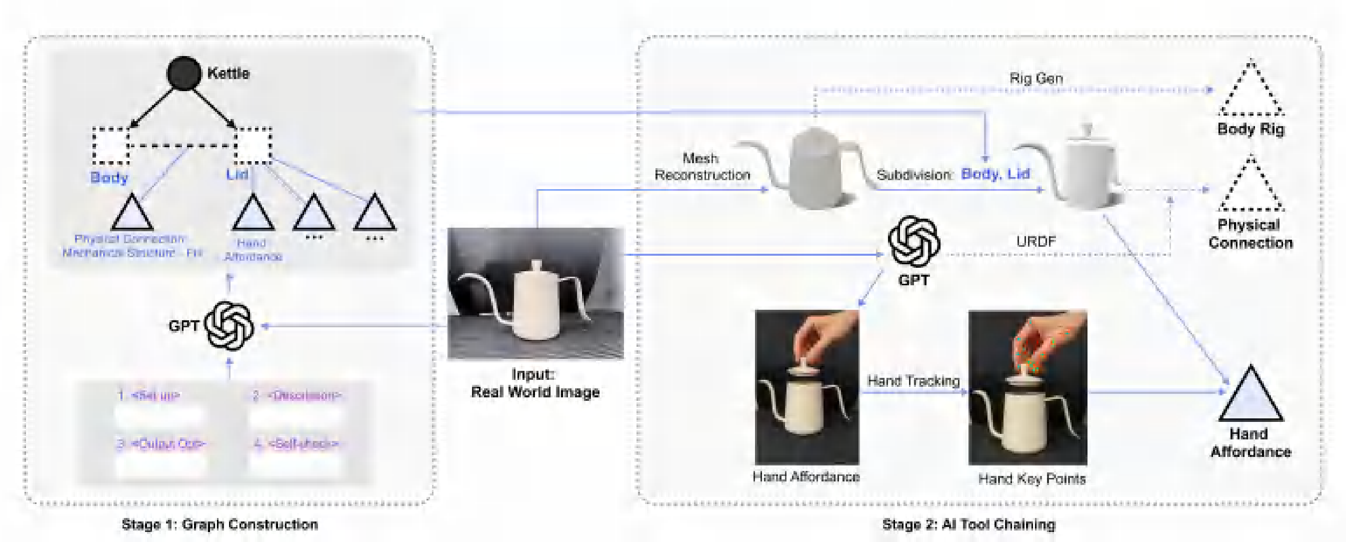
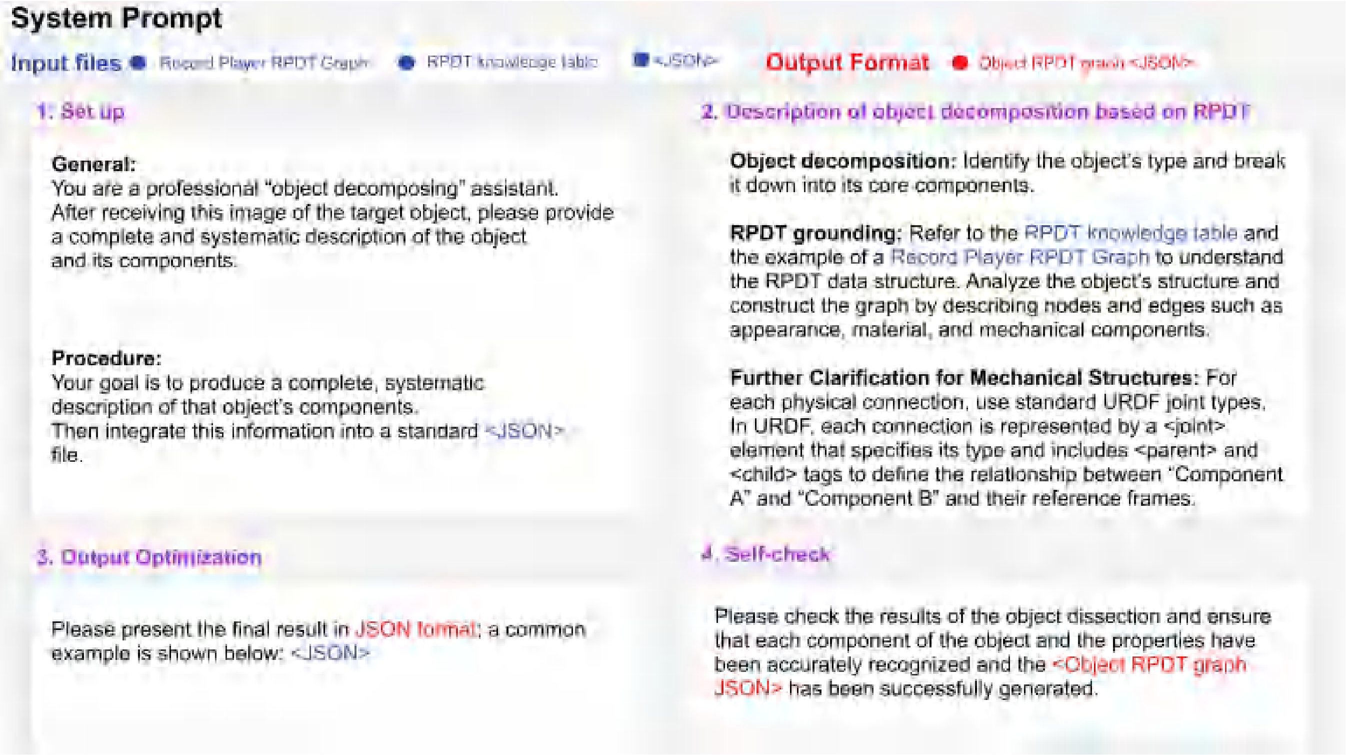


Figure 2: Workflow. The workflow of RealTwin consists of two stages. Stage 1 performs graph construction driven by the MLLM, and Stage 2 uses the AI tool chaining to realize the kettle's attributes informed by the graph. Dashed lines indicate the workflow for other attributes in a standard setting.



System Prompt

Input files ● Record Player RPDT Graph ● RPDT knowledge table ■ <JSON> Output Format ● Object RPDT graph <JSON>

1. Set up

General:
You are a professional "object decomposing" assistant. After receiving this image of the target object, please provide a complete and systematic description of the object and its components.

Procedure:
Your goal is to produce a complete, systematic description of that object's components. Then integrate this information into a standard <JSON> file.

2. Description of object decomposition based on RPDT

Object decomposition: Identify the object's type and break it down into its core components.

RPDT grounding: Refer to the RPDT knowledge table and the example of a Record Player RPDT Graph to understand the RPDT data structure. Analyze the object's structure and construct the graph by describing nodes and edges such as appearance, material, and mechanical components.

Further Clarification for Mechanical Structures: For each physical connection, use standard URDF joint types. In URDF, each connection is represented by a <joint> element that specifies its type and includes <parent> and <child> tags to define the relationship between "Component A" and "Component B" and their reference frames.

3. Output Optimization

Please present the final result in JSON format; a common example is shown below: <JSON>

4. Self-check

Please check the results of the object dissection and ensure that each component of the object and the properties have been accurately recognized and the <Object RPDT graph JSON> has been successfully generated.

Figure 3: Prompt design of RPDT graph construction.

MLLM (e.g., GPT-4o[2]), which is prompted to leverage its image-generation/editing capability to iteratively annotate joint positions directly on the input image, returning the same image with the 2D joint position marked in red dots. For each physical connection edge, a structured URDF-based prompt specifies the joint type

[2] https://openai.com/

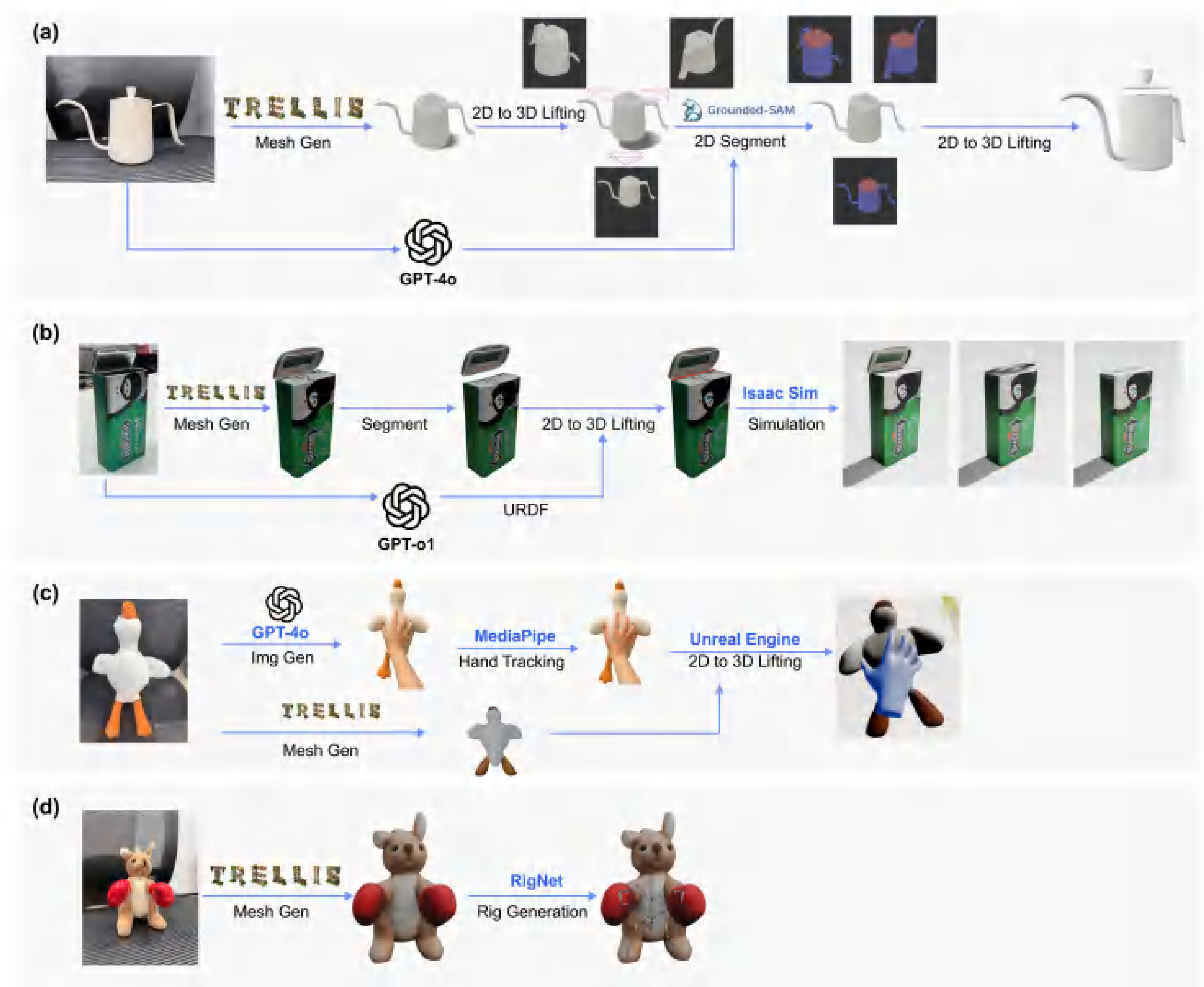


Figure 4: Tool Chaining to realize attributes: (a) Mesh reconstruction and subdivision; (b) Mechanical structure reconstruction; (c) Hand affordance generation; (d) Body rigging.

along with its joint type and constraint descriptions. By synthesizing 2D annotations from multiple views and lifting them to 3D collectively, the 3D joint position can be acquired. This process can be formulated as an optimization problem:

$$p^* = \mathbf{Proj}(argmin_p \Sigma_i ||\mathbf{OrthProj}(C_i, p) - x_i||^2, M) \qquad (2)$$

where p is the 3D point to be optimized; x_i is the 2D annotation given by the MLLM for the image captured by the i^{th} camera. The **Proj** function means projecting the optimized point to the object mesh as the joint position. Although two non-collinear views are theoretically sufficient for triangulating a 3D joint, relying solely on two views proved fragile in practice due to annotation noise, object self-occlusion, and symmetric geometries commonly found in household articulated objects. To balance robustness and annotation cost, we adopt three evenly spaced orthographic views. After acquiring the joint position, the joint's rotation can be estimated by: 1) fitting the set of split points with a plane P_1, getting a perpendicular unit vector x_1; 2) intersecting P_1 with the mesh's tangent plane at the joint point to get a unit vector x_2; and 3) getting an orthogonal coordinate system $(x_1, x_2, x_1 \times x_2)$ to represent the rotation. Meanwhile, the joint's constraints can be retrieved based on the MLLM's prior knowledge of the object and the joint type.

A3: Hand affordance generation. Hand affordance means how the hand can interact with an object. To automatically generate a set of plausible affordances for an object, the system first generates textual gesture descriptions that illustrate the feasible hand gestures for interacting with the object using MLLM. In this step, the MLLM is prompted to first consider the object's identity and typical usage, and then reason about which hand–object interactions would be plausible given these functionalities, before listing them. The instruction is: "Considering the object's name and functionalities, think about what hand gestures are plausibly involved in using this object, and list as many such gestures as possible, describing each in detail." Despite this procedure, the MLLM may still hallucinate and produce implausible gestures. However, as described in Sec. 3.3.3, human users can easily correct such cases through a simple demonstration-based user interface in AR environment. Leveraging the MLLM's image generation capability, a prompt with the format *"[img] Given this object, generate an image with a hand interacting with the object in the following manner: [gesture description]. Keep the object shape and position unchanged in the output image."* is used to generate an image with the hand affordance. Then, a hand tracking model (e.g., MediaPipe) is applied to detect the 2.5D hand pose (e.g., relative to the root in depth) from the image. In order to get the hand's transform relative to the object's mesh, we use TRELLIS to simultaneously reconstruct the 3D meshes of both the image only with the object and the image with the object and the interacting hand. Then we align these two meshes and choose one keypoint that is not occluded as the "reference keypoint". By projecting the reference point to the mesh with the interacting hand, and updating the rest of the keypoints by relative position from the pose, the hand skeleton can be aligned to the object mesh, as is shown in Fig. 4(c).

A4: Body rigging. For body rigging, we directly apply RigNet [90], a model that can automatically generate rigging from the 3D mesh, on the reconstructed mesh on all the components with the "rigging" attribute, as is shown in Fig. 4(d).

Throughout the above AI tool chaining approach, we effectively infer complex attribute values such as 3D meshes, mechanical structures, hand affordances, and body rigs by integrating various AI models with MLLM inference, thereby enriching the object representation and enabling more sophisticated downstream applications.

3.3.3 End-user Interfaces. Building on the automatic grounding framework, we design a lightweight user interface that enables users to rapidly construct an RPDT from simple text or image inputs, with minimal learning and usage cost. Through this interface, users can directly edit intermediate results within the grounding framework, thereby maintaining the freedom to apply personalized modifications or correct errors made by the AI tools, as detailed below.

1) Graph element correction through natural language. In **S1**, when the generated concept graph contained errors, users were able to revise it by issuing natural language commands (e.g., "the connection between the handle and the body is wrong; it should be a fixed connection").

2) Interactive refinement of segmentation. In **A1**, when the Grounded-SAM segmentation produced errors, users could refine the result by interactively adding positive points (regions expected to be inside the mask) and negative points (regions expected to be outside the mask) on the image, as is shown in Fig. 6. The system then updated the mask accordingly and provided real-time feedback, enabling users to iteratively adjust the segmentation results.

3) Hand affordance demonstration. In **A3**, users may first review the MLLM-generated hand affordances to assess their plausibility. If an affordance is incorrect or if additional affordances are needed, users can enter the AR editing environment, where the object's 3D model is preloaded. The virtual object can be freely positioned in space to a convenient location. Once placed, the user presses a button to begin recording and directly performs the intended gesture on or around the virtual object. During this process, the system captures all hand keypoints and the hand–object relative pose, and binds the demonstrated gesture to the corresponding object. A second button press finalizes and saves the new gesture. Once bound, repeating the same gesture near the object in AR is recognized by the system, thereby enabling user-defined hand affordance creation, as is shown in Fig. 7. The virtual object becomes an interactive 3D asset in the AR environment, allowing users or developers to directly manipulate it or further script its behaviors based on the newly defined affordance.

4 Evaluation

This section presents an evaluation on the performance of 1) MLLM-driven graph construction for RPDT, including the inference accuracy of the nodes and edges for each object, and 2) the success rate of attribute values inferred by MLLM and realized by the AI tool chaining.

4.1 Evaluation of MLLM Graph Construction

4.1.1 Design and Metrics. To evaluate the accuracy of MLLMs in constructing the RPDT concept graph, we selected 50 objects for testing. To ensure that each object's graph contains at least two sub-vertices and one edge, we prioritize objects with articulated structures or those that have more than one component. We include 10 articulated objects from the PARIS dataset [61], which are rendered using Blender to synthesize the textured images as the input into our test set. We further include 40 daily objects from 9 categories with real-world image captures: 11 containers, 5 stationery items, 5 bathroom products, 4 home appliances, 4 small tools, 4 pieces of furniture, 3 game controllers, 2 kitchenware items, and 2 pieces of fitness equipment. The test set constitutes a total of 50 images for reconstruction. Two experimenters independently evaluated all results requiring human assessment, after which they discussed and resolved any disagreements to finalize the results.

We used GPT-o1 [3] as the base MLLM model, which supports multimodality (text+image) inference for both input and output, to construct the concept graphs with the prompt shown in Sec. 3.3.1. We choose o1 for its strong multimodal grounding and instruction following, reliable structured outputs (e.g., JSON) needed for graph parsing, and stable long-context performance observed in our pilot tests. During the construction, some attributes that can be described through text are also inferred, which are also evaluated these attributes in this section. The attribute keys that we included in our evaluations are component description and material type for the vertices.

For each object, the experimenters provide a ground-truth concept graph $G_fRPDT = (V_f, E_f)$ with the abovementioned textual attributes, where the number of non-root vertices is denoted as N, For each concept graph generated by the MLLM $G_{RPDT} = (V, E)$, the number of vertices that match the ground-truth concept graph

[3] https://openai.com/o1/

is denoted as N_a (matching means that the description of each attribute of the vertex is consistent), and the number of redundant vertices generated is denoted as N_b. The inference recall $\frac{N_a}{N}$, precision $\frac{N_a}{N_b+N_a}$, and F-1 score $\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$ are reported. Similarly, we also evaluate the edges for each object graph and calculate their recall, precision, and F-1 score. For the attributes of component description and material type, the experimenter manually examines the correctness of the text.

4.1.2 Results. We report the overall results in the Table 2. The main failure in vertex prediction occurs with more complex objects (such as fitness equipment with more than 10 components). These objects often contain many small parts and internal structures that are difficult to observe from images, which can be overlooked by the MLLM. Additionally, since MLLMs infer structures and components based on common knowledge bases, they may struggle with customized or modified items. For example, if an Xbox controller has customized buttons covered with protective caps, the MLLM may not accurately interpret their structure.

As for the edges connected by vertices, the main issue lies in multi-degree-of-freedom mechanical structures that are difficult to infer. Take the stylus of a vinyl record player as an example—it can rotate around its base and can also be lifted upward by the finger to reach the record, meaning it has an additional degree of vertical freedom. The MLLM initially missed this detail during the first round of graph construction. However, when the experimenter provided a second round of decomposition prompting (e.g., "Could you check the movement style of the stylus?" and requested an updated graph), the MLLM constructed a completely accurate graph.

Table 2: Results for the graph construction

	Recall	Precision	F1 score
vertices	99.3%	98.0%	98.3%
edges	91.3%	95.0%	92.2%
component description	95.4%	95.8%	95.0%
material type	93.8%	96.7%	94.4%

4.2 Evaluation of Attribute Value Inference

Further, we evaluate how graph attributes are inferred on the constructed concept graph with MLLM and AI tools, using GPT-4o[4] as the backbone model for its strong image-generation consistency—advantages such as stable style control, coherent multi-element composition, and high-fidelity texture rendering—supporting more reliable visual attribute inference. All of the attributes described in Sec 3.3.2, except body rigging, are evaluated. This is because body rigging can be independently implemented by the external RigNet [90] without the involvement of MLLM and any algorithms developed within RealTwin. In addition, rig generation is mainly relevant for deformable objects (e.g., animals, plants, soft-body structures). We evaluated the rigging performance on 10 animal-like furry toys and obtained the following results: recall = 0.914, precision = 0.749, and F1 = 0.763. These metrics were computed using the number of ground-truth rigs (N), the number of correctly identified rigs (N_1), and the number of falsely predicted extra rigs (N_2), following: recall $\frac{N_1}{N}$, precision $\frac{N_1}{N_1+N_2}$, and F-1 score $\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$. Further discussion on the future development of body rigging is provided in Sec. 6. Two experimenters independently evaluate all results requiring human assessment, after which they discuss and resolve any disagreements to finalize the results.

4.2.1 Mesh reconstruction and subdivision. We randomly selected 30 objects from a real-world object set in Sec 4.1.1 to examine the quality and accuracy of mesh reconstruction and subdivision. For each object, our experimenter captured 3 images of the object in the real world from different views. The TRELLIS [86] model was employed to generate the 3D mesh of the object. After obtaining the 3D mesh, for each child component of this object, we select $K = 3$ to render the orthogonal projection maps I_1, I_2, I_3, using a distributed camera setup. The distributed camera C_1, C_2, C_3 is fixed at an elevation angle of 36 degrees relative to the 3D mesh, rotating 120 degrees in azimuth around the mesh each time to render 3 orthogonal projection maps. Next, Grounded-SAM [76] was used to generate the segmentation masks based on the child component's text description for every projection map, followed by applying Formula 2 to generate the mesh segment that belongs to this child component. If a component's segmented mesh was visually correct and consistent with its description, the experimenters would label it as successfully segmented.

Results: These 30 objects have a total of 72 components to be subdivided, with 60 components successfully subdivided, resulting in an acceptance rate of 83.3%. Beyond the overall success rate, the failure cases reveal several reconstruction challenges. Most errors arise when components are occluded or internally hidden, such as ink refills or inner liquids, making them visually inaccessible for projection-based segmentation. Transparent or reflective materials also create ambiguous boundaries, and fine mechanical seams are often too subtle to preserve across multi-view projections, leading to merged or incomplete subdivisions. These challenges can be partially mitigated through our human-in-the-loop interface, since users inherently understand the semantic structure of the object and can correct missing components or refine boundaries through interactive edits. For the long-term goal of fully automatic and large-scale RPDT construction, future developments of RealTwin may need to integrate more advanced 3D generation models that incorporate fine-grained structural semantics of the object, or the object concept graph that we design, as prompts to help the generation process.

4.2.2 Mechanical structure reconstruction. We randomly selected 30 objects from a real-world object set in Sec 4.1.1 to assess their mechanical structural attributes inferred by MLLM. For each joint connecting two components, after getting the component 3D mesh segments from the previous step in Sec. 4.2.1, we use the URDF format to represent the joint between the segments. URDF describes a joint connecting two meshes by defining a joint with its type, parent link (parent mesh segment that remains stationary or fixed in the object's frame of reference), child link (child mesh segment that moves relative to the parent link), joint position, and the range of motion constrained by the joint, establishing the motion relationship between the parent and child links. We implemented the

[4] https://openai.com/index/hello-gpt-4o/

robot simulation environment Isaac Sim [5] to read and visualize URDF files, simulating the jointed mesh segments' motions. The experimenter could observe the simulated motions defined by the URDF file in Isaac Sim to evaluate whether the motion is consistent with the real-world objects. The key parameters to define a joint are joint type, joint position, and range of joint's motion. We evaluated the ability of MLLM to infer these three parameters. The joint type and range of motion can be described using language, so these two parameters can be directly generated by MLLM. Additionally, we have specified that GPT's output format adheres to the URDF standard description. The joint position is a 3D coordinate, which MLLM cannot provide directly. Therefore, we used the method mentioned in Sec. 3.3.2, utilizing GPT-4o's image generation feature to annotate the 2D orthogonal camera render maps in Sec. 4.2.1, and then project the annotations onto the original 3D mesh to get the joint position. The experimenter then imports the URDF file into Isaac Sim to observe whether these parameters are correctly defined and records the acceptance for three parameters, respectively.

Results: The 30 objects collectively contain 35 joints. The acceptance for joint type prediction reached 97.1%, and the acceptance for joint limit prediction reached 94.2%. For joint position prediction, the acceptance of the MLLM annotations was 88.7%. A total of 26 joints were successfully predicted and simulated in Isaac Sim with all three parameters correctly predicted. Beyond the overall acceptance rates, the failure cases reveal that most errors occur not in identifying the correct joint type, which the MLLM often predicts reliably, but in locating the precise joint position on the 3D mesh. When the visual cues in the rendered projections do not clearly expose where two components physically connect, the model tends to place the joint at a visually plausible but incorrect region, such as the opposite side of a lid or at the center of a symmetric body. This issue becomes more pronounced for objects with enclosed hinges or joints embedded within thick shells, where the true articulation point is only visible from specific viewpoints. Symmetric geometries further amplify this ambiguity because multiple positions appear equally reasonable from the projections alone. Another source of failure arises in objects with tightly coupled or multi-joint assemblies, where the relative motion of adjacent components cannot be distinguished from isolated single-view observations, causing the system to collapse multiple degrees of freedom into a single simplified joint. These observations suggest that accurate joint generation requires additional constraints beyond appearance, such as enforcing geometric consistency across views, incorporating physical plausibility checks on candidate motions, or leveraging structural semantics from the concept graph to restrict the feasible joint inferences. Future extensions may also incorporate lightweight physical reasoning modules that test the plausibility of a predicted joint by simulating candidate motions, or integrate motion exemplars from internet videos to ground the model's understanding of how similar mechanisms behave.

4.2.3 Hand Affordance Generation. We randomly selected 30 objects from the real-world object set in Sec 4.1.1 to evaluate hand affordance prediction using MLLM. For each component of the object, the experimenters first provide MLLM with a photo of the real object. Then they use GPT-4o to generate photos that demonstrate the interactions between hands and this target object. The experimenters then assess the generated images, evaluating whether they are physically plausible, labeling them as True or False. Finally, they discuss and finalize the result for each image.

Results: A total of 129 hand affordance images were generated for the 30 objects, with an accuracy of 92.5%. The failure cases highlight inherent challenges in generating reliable hand–object interactions purely from static images and language prompts. Most errors arise when the MLLM infers a plausible but physically incorrect gesture due to limited visibility of graspable regions, leading to hand poses that either intersect the geometry or fail to align with the component's true affordance. Additional failures occur when the object containing joints is misinterpreted, for example, mistaking a hinged lid for a detachable one, which leads the system to generate removal-like gestures instead of rotational manipulations. Potential extensions include integrating contact region detection to ensure that generated hand poses align with feasible grasp surfaces, employing interaction plausibility checks that evaluate whether forces, orientations, and contact points are physically reasonable, and constraining hand poses based on the object's structural semantics to rule out invalid manipulations. Future developments may also leverage small-scale hand–object simulation modules or hand-object motion examples to better align generated gestures with realistic human interaction patterns.

4.2.4 Processing Time and Hardware Implementation. We report execution time using a workstation equipped with an NVIDIA H20 GPU (96 GB VRAM, CUDA 12.4), an AMD EPYC 9K84 96-Core Processor (16 vCPUs allocated), and 153 GB of system memory. To compare how object complexity influences computational cost, we distinguish between simple objects and complex objects: an object is considered complex if its concept graph contains more than two leaf nodes or more than three non-leaf nodes.

The full pipeline requires **153.3 ± 18.6** s for simple objects and **399.0 ± 27.9** s for complex objects on average. The results show that simple objects maintain relatively stable and moderate latency across stages, reflecting the lower structural ambiguity and limited graph depth involved. Complex objects show increases in processing time, particularly in stages involving MLLM-based reasoning and rig generation. These stages incur additional computational overhead due to the larger and more intricate object graphs, denser structural relationships, and the need to infer and construct more partial physical properties. Even with differences, the overall pipeline remains tractable for reconstructing RPDTs.

5 User Study

To evaluate the practical applicability of our framework, we conduct a user study to understand participants' experience with RealTwin in practice. To further explore broader usage scenarios with design implications of RPDT, we recruit participants from diverse professional backgrounds, contributing ideas on new extensions for RPDTs in the future.

5.1 Participants and Procedure

We recruit 12 participants through questionnaires from the campus, aiming for a diverse mix of professional backgrounds (6 male, 6 female; age: avg = 26.1, std = 2.9). The participants include 1 VR

[5] Isaac Sim: docs.isaacsim.omniverse.nvidia.com/latest/index.html

developer, 2 installation artists, 2 architects, 1 fashion designer, 2 product managers, 2 game developers, 1 curator, and 1 industrial designer.

We use a pour-over kettle as the example object. Guided by an experimenter, participants are first introduced to the concept of RPDT and the workflow of the framework. Participants then capture photos of the object, from which the framework automatically generates a concept graph and corresponding attributes using AI tools. Participants observe both intermediate outputs (e.g., graph structure and segmentation results) and final attributes in interactive environments, such as joint movements in Isaac Sim and hand affordance recognition in VR. When the framework inference errors occur, participants use the interfaces described in Sec. 3.3.3 to refine intermediate results; otherwise, prepared error cases are provided to let them experience the editing process. Throughout the study, participants are encouraged to think aloud and customize edits via the interface. Our study also included a free-exploration phase, during which participants were encouraged to try reconstructing any additional objects available in the environment. At the end, each participant completes a semi-structured interview (see Appendix A for full questions) and a seven-question Likert-scale questionnaire evaluating their experience with the framework. The interviews are video-recorded by two experimenters. The study was conducted in our laboratory and received ethical approval from the university's ethics review board.

5.2 Results

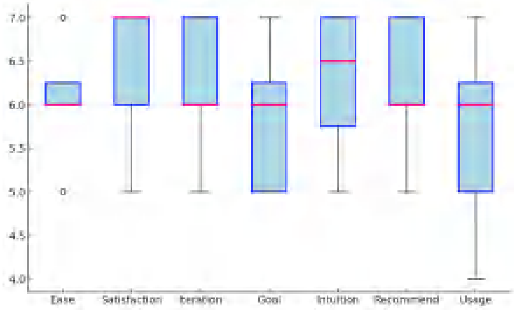


Figure 5: Distribution of subjective rating scores of the questionnaire for user experience. 1 -strongly disagree, 7 - strongly agree.

We employed a mixed-methods approach in analyzing the user study. The questionnaire responses provided a broad overview of participants' perceptions. Qualitative data from the semi-structured interviews were analyzed using thematic analysis. In total, we collected approximately 12 hours of video recordings from 12 participants. All recordings were transcribed using a commercial ASR service (iFLYTEK[6]) and subsequently reviewed for accuracy by two experimenters. Then, the experimenters independently conducted open coding on the transcripts. Coding disagreements were discussed between the two experimenters until a consensus was reached. Based on the final coding results, the whole research team worked together to iteratively group the codes and identify recurring themes, which are used to organize our findings. The combined insights informed a set of design implications, which are presented in the following subsections.

As shown in Fig. 5, the questionnaire results indicated that participants generally reported positive experiences with the framework, particularly in terms of satisfaction with the outcome, perceiving the workflow as intuitive, and enabling iterative refinement. The responses also revealed opportunities for improvement, especially in helping participants achieve their intended goals and in supporting broader adoption in future use.

5.2.1 *Intuitive and Engaging Reconstruction Experience by RealTwin.*

Participants consistently described RealTwin as both easy to comprehend and open to active user participation. Unlike opaque black-box systems, RealTwin's step-by-step process—generating attribute graphs and chaining AI tools—was perceived as transparent and aligned with users' own reasoning about object structures (P7, P10). For instance, P10, a designer with professional product modeling experience, explained that *"traditional workflows usually begin by modeling components and then assembling them, a process that requires substantial prior expertise"*. By contrast, RealTwin approaches reconstruction by decomposing a whole object, which P10 noted *"felt more consistent with human intuition about how physical structures are understood"*. This comprehensibility lowered the entry barrier and made the workflow approachable even for those without deep 3D modeling expertise. At the same time, participants valued the opportunity to directly and actively participate the reconstruction process. Through using interactive interfaces, participants were able to inject their own understanding and intent into the pipeline (P1, P7, P9). As one installation artist highlighted, *"If I want to add special interactions to my artwork, I can simply demonstrate them myself, and they can be directly bound to the digital twin of the piece—this is incredibly convenient."*. This participatory element not only improved the reliability of the outcomes but also fostered a stronger sense of collaboration and co-construction with the framework (P8, P11).

5.2.2 *Efficient Workflow and Richness in Concepts of Digital Twin.*

When reflecting on their prior experiences with 3D modeling and reconstruction tools such as Blender, Rhino, Unity, and Reality Composer, participants highlighted a clear contrast between traditional workflows and RealTwin. Conventional methods were described as time-consuming, rigid, and requiring significant manual effort. For example, when using traditional 3D scanning applications, participants noted that the workflow typically involved holding a camera or mobile phone and moving around the object, or even flipping it, to capture sufficient geometric information from multiple angles (P7, P12). By contrast, RealTwin's automated pipeline of attribute graph generation and AI tool chaining reduced the need for repetitive manual operations (P7, P11). Participants also appreciated the richness of the RPDT concept. They noted that while traditional pipelines mainly focus on reconstructing static models, RPDT makes it possible to reconstruct additional properties of physical entities, such as physics and affordances (P1, P3, P4, P10). Because these properties can be represented through a relatively easy process, participants noted that the resulting digital twins appeared more realistic and were easier to adopt in virtual environments (P4, P9, P10, P11). For

[6] https://www.iflyrec.com/zhuanwenzi.html

example, this was considered particularly valuable in game development, where *"many virtual interactions are inspired by and aligned with those of the physical world"* (P10). Similarly, in the fashion design industry, the participant highlighted that RealTwin could *"quickly reconstruct how different fabrics and cuts would look on a model, greatly reducing both cost and time"* (P12).

5.2.3 Challenges and Implications in Reconstruction Fidelity and Structural Complexity. Participants also identified several challenges and future design implications for the framework. The most frequently mentioned concern was the difficulty of reconstructing internal or fine-grained structures that are not directly visible, such as the liquid inside a cup, the layered compartments of a refrigerator, or the cavities of centrifuges and other delicate mechanical components (P1, P10, P3). Some suggested that the framework could include predictive modeling of internal structures while preserving interfaces that allow users to make manual corrections, which would improve the accuracy of container-like objects. Participants also noted that the current framework struggles with complex mechanical systems, such as assembly line machinery or weaponry, where fine-grained functional details are critical (P3, P10). Others mentioned limitations in handling large or fragile objects when demonstrations were required (P8). For tasks demanding industrial precision or rigid control, participants suggested that traditional CAD or simulation tools might still be preferable (P4, P8, P11). For example, one participant described the idea of creating a digital twin of a fishing rod and recording the action of lifting a fish out of the water as part of the rod's affordance. *"This action involves cun jin, a subtle wrist motion that generates precise force within a very short range"*, which participants noted might be too delicate for the current framework to faithfully capture (P10). Finally, they emphasized the need to balance ease of input with fidelity of results, observing that overly simplified descriptions could lead to incomplete or oversimplified reconstructions (P6, P8).

6 Discussion

This discussion examines the technical evaluation results, the potential for large-scale scalability, and the user-centered insights from our study. We highlight the strengths and current limitations of MLLM-based grounding, explore how large-scale RPDT generation could stimulate future AI-driven innovation, and envision how active end-user participation can enable personalized, creative RPDT reconstruction and help cultivate a thriving RPDT ecosystem.

6.1 Implications on MLLM-based Grounding for RPDT

In this paper, we implement an MLLM-based RPDT grounding framework with a systematic evaluation on MLLM's capability as the backbone to drive the reconstruction of objects in the physical world for the first time. Taking advantage of the cross-modality prior knowledge of the physical world that MLLM has learned from, our results have verified that MLLM generally has a good zero-shot grounding capability for real-world objects in terms of appearance, physics, functionality, and interactivity, achieving a consistently high F-1 score for concept graph construction (98.3% and 92.2%), material recognition (94.4%), mechanical structure recognition (97.1%), and hand affordance inference (92.5%). We also found that such real-world grounding capability can be further enhanced when MLLM is equipped with a vast AI toolset, allowing MLLM to overshoot its inherent capability limitation (e.g., 3D generation) by chaining AI tools. The mutual development of MLLM and AI tools has also formed a good complementary effect to push forward the boundary of physical world grounding.

Meanwhile, drawing from the failure cases in our evaluation, we find that MLLM is still hard to deal with situations where the object to reconstruct is out of the MLLM's knowledge scope, extremely complex or precise in the structure, or yielding ambiguities (e.g., invisible structure, specific materials) for grounding. It's an essential question how to address these challenges to make RPDT grounding even "more realistic". A potential solution is to optimize MLLM's knowledge base, knowledge updating mechanism, and instant knowledge acquisition strategy. For example, when dealing with an object with complex mechanical structures, the user can upload the documentation of this object to provide MLLM with additional details. When dealing with unseen objects or ambiguities, MLLM can actively seek help from the user (e.g., requiring them to provide additional information or clarification), where the user's response can also be used to fine-tune the model [13, 40, 95]. Future research on these topics is key to enhancing the robustness and adaptability of RPDT grounding, making it more capable of handling diverse and complex real-world scenarios.

While the current framework contains multiple interdependent components, this modular design is necessary because each stage isolates and resolves a distinct class of failures that single-stage, end-to-end approaches cannot reliably handle. Decomposing the process into attribute-graph construction, geometric reconstruction, mechanical-structure inference, affordance generation, and rig synthesis produces explicit, standardized intermediate representations such as part-level meshes, URDF-like joint descriptions, and VR-ready interactive assets. These representations make the reasoning process interpretable, enable human users to inspect and correct outputs at each step, and preserve fine-grained control over structural and interaction fidelity, which would be unattainable in a monolithic black-box model.

Moreover, much of the framework computational cost stems from the performance of deployed AI modules (e.g., Trellis [86] for mesh reconstruction, RigNet for rig generation [90], SAM2 [76] for segmentation). These domain-specific modules can be replaced directly as more capable models emerge, improving the framework's performance naturally with advances in domain-specific AI models. For example, faster mesh reconstruction, more reliable 3D subdivisions, or more accurate and lightweight rigging methods could directly reduce latency. Moreover, the framework can be simplified in certain user scenarios. For example, in a home-placement or interior-decoration simulation AR application, users may only require approximate geometry and basic affordances for positioning and visualization. In such cases, detailed rig synthesis and component-level physical property inference are less critical, and the pipeline can safely skip these modules to significantly improve runtime performance.

6.2 Scaling RPDT for Large-Scale Data Mining and AI-Driven Innovation

We propose a concept graph to represent RPDT, enabling the unified decoupling of object reality properties. Each vertex and edge in the graph is computable and interpretable. Integrated with our grounding framework of MLLM and AI tools, this graph can serve as the object's recognizable blueprint by AI, enabling large-scale mining of internet data to generate a vast amount of product blueprints. As MLLMs and AI tools rapidly evolve, more and more internet data could be used to contribute to the automatic generation of blueprints. One promising direction is the product images and videos on the internet. For example, product review videos typically include interactions between the blogger and the product, such as testing all the product's features and presenting the results. In this case, MLLM can analyze the video to identify the product's possible affordances and interactivity, product durability, or how the object behaves under different conditions, thereby enriching the construction of the product's digital twin blueprint. This ability to extract nuanced information from a variety of content sources leads to highly detailed and accurate digital twins, surpassing traditional methods.

Moreover, as the scale of RPDT generation expands, AI could evolve from merely reconstructing objects to creating entirely new ones, much like an inventor. By training AI systems on extensive RPDT datasets, these systems could autonomously generate new product designs and blueprints. This capability could herald an era where AI-driven innovation not only replicates existing designs but also invents new products based on a profound understanding of physical properties and human interactions. In this context, scaling up RPDT generation could lead to the emergence of novel AI-driven designs and manufacturing, pushing the limits of digital twins. This is essential for embodied intelligence, as scaling up digital twins enables AI systems to better comprehend and interact with the physical world, creating new opportunities across industries and transforming how we design, interact with, and innovate in both virtual and physical environments. With reinforcement learning or online learning techniques [13], MLLMs could continuously refine their understanding, adapt to specific reconstruction needs, and improve the accuracy and efficiency of the models over time. This process would allow for more personalized and context-specific digital twins, making the system more adaptable and responsive to real-world changes.

As the metaverse continues to evolve, the demand for automated, real-time, and in-situ digital-twin generation is increasing. As a general and extensible digital-twin reconstruction pipeline, RealTwin still requires several key advancements to move toward real-time performance. In VR/AR usage scenarios, users are inherently equipped with headset-mounted cameras that continuously capture the geometry and appearance of surrounding objects while providing stronger contextual awareness during real-time interaction. Future versions of the system could leverage this capability to reconstruct objects in a streaming manner [53, 55], rather than relying solely on offline, multi-stage processing. First, essential stages of the pipeline must evolve toward single-pass or near-real-time inference, including high-speed multi-view fusion, low-latency geometry reconstruction, and direct articulation prediction from sparse views [8, 53, 55]. Second, affordance inference and rig generation must support incremental, streaming-based updates, avoiding full regeneration whenever the object or its viewpoint changes [54, 73]. Third, attribute-graph inference will require high-throughput MLLMs or locally distilled lightweight models to reduce multimodal round-trip latency [24, 57]. Finally, tighter integration with on-device sensing and spatial tracking (e.g., depth sensing, hand tracking) would allow the system to progressively refine the digital twin as the user moves around the object.

6.3 End-User-Oriented RPDT: Ecosystem, Personalization, and Creative Potential

Our user study revealed that RealTwin enables end users to reconstruct digital twins in ways that are intuitive, engaging, and efficient. Participants emphasized its transparency and step-by-step process, which aligned with human reasoning about object structures, while also lowering the entry barrier compared to traditional 3D modeling tools. They valued the opportunity to actively participate in the reconstruction process through interactive interfaces, enabling them to inject their own understanding and intent. However, they also highlighted challenges in reconstruction fidelity and structural complexity, particularly in capturing internal details and intricate mechanical systems, pointing to the importance of balancing ease of input with reconstruction accuracy.

The framework we propose for reconstructing RPDTs is therefore not only fully automated but also incorporates interactive interfaces that allow users to refine intermediate results. This combination empowers individuals without specialized expertise to reconstruct objects with high precision while still supporting customization and adaptation to diverse usage contexts. As a result, we envision that the framework could contribute to the rapid growth of digital assets on the internet, sparking creativity in a manner similar to AI-driven image generation technologies, which have enabled the general public to become digital creators.

Although RealTwin already supports user-driven modifications of intermediate reconstruction results, we propose the development of a more accessible framework that further improves usability. Beyond the current interactive interfaces, several additional capabilities are needed, as highlighted by our findings. First, to address the challenge of reconstructing internal or fine-grained structures, the system should support predictive modeling combined with user correction to balance automation with fidelity. Second, to handle complex mechanical systems, the framework should provide modular editing tools that allow users to refine joint types, motion ranges, or dependencies with greater precision [65]. Third, considering large or fragile objects, lightweight or remote demonstration mechanisms (e.g., adaptive gesture simulation in AR without physical manipulation) could expand applicability across more contexts [100]. Finally, to reduce the risk of oversimplified outputs, usability could be enhanced by multi-step guidance or visual feedback that helps users iteratively refine reconstruction without requiring deep expertise. Previous work has also highlighted the significance and challenges of personalized digital twin reconstruction [60], which necessitates that the digital twin incorporates more personalized elements, such as usage marks on objects or handcrafted decorations. Existing research has demonstrated the potential of using natural language to modify the attributes in our RPDT graph (e.g., materials [98], textures [16, 62], and 3D models [17, 18, 66, 92]). By integrating these advancements into our framework, RealTwin can be more accessible to end-users to reconstruct objects with personalized features or customized modifications.

Looking ahead, we envision that combining fully automated reconstruction with user-driven participation will be essential for the next generation of RPDT frameworks. Such hybrid systems could automatically scale RPDT generation across massive datasets while preserving avenues for end-user interventions to inject personal, contextual, or domain-specific knowledge. This balance between automation and human-in-the-loop participation will not only enhance usability and fidelity but also foster broader adoption across industries, shaping an ecosystem where RPDTs become both universally accessible and deeply customizable.

7 Limitations and Future Work

RealTwin supports the reconstruction of common affordance interactions, but more diverse forms of real-world human–object affordances should also be included. For example, indirect interactions (e.g., pressing a button to open a lid), or physical–digital interaction mappings involving electronic components (e.g., pressing the spindle of a record player should trigger music playback). Prior work has explored how users may author such complex interactions in AR environments [60, 100]; however, scaling up the automatic construction of these attributes with logic-driven affordances remains an open challenge, requiring more expressive interaction grammars, richer causal models, and improved physical–digital correspondence inference.

In addition, RealTwin does not yet support continuous hand–object interactions or multi-hand gesture bindings. For example, continuously squeezing a ketchup bottle to extrude sauce, or holding an Xbox controller with two hands while pressing buttons with different fingers. These interaction processes involve time-dependent state transitions, continuous deformation tracking, and complex conditional logic, which our present pipeline cannot capture. We plan to focus future work on modeling and binding these temporally extended and multi-limb interaction behaviors.

The complex coupling among physical attributes should also be considered in a more fine-grained digital-twin reconstruction. Also, attributes such as material, mass, or mechanical constraints directly affect interaction dynamics. For instance, objects with identical appearance but different weights will exhibit different lifting motions, and interactions between hands and special surface materials can cause noticeable appearance changes (e.g., a soft balloon deforming when squeezed). These factors represent important directions for extending the realism and fidelity of the framework.

Furthermore, beyond inanimate objects, the lightweight, low-barrier construction of digital twins for living entities (e.g., pets, people, or moving agents) is another important future direction. Interacting with the digital twin of a living creature requires modeling autonomy, behavioral dynamics, and safety constraints.

8 Conclusion

In this paper, we introduce the concept of the Reality-Preserving Digital Twin (RPDT) with a scalable concept graph representation, derived from a human-centric perspective. We propose an MLLM-based grounding framework with AI tool chaining for RPDT reconstruction. The technical evaluation showcases the MLLM's ability to reconstruct physical-world objects by utilizing its cross-modal knowledge. Our user study reveals that RealTwin is intuitive, engaging, and efficient, and participants valued its transparent step-by-step process, which mirrors human reasoning about object structures and lowers the learning curve for non-experts. Enlightened by RealTwin, we discuss critical issues, such as future digital twin ecology, interaction space, and real-world adoption toward truly end-to-end fine-grained and scalable digital twin reconstruction.

Acknowledgments

This research is partially funded by the Natural Science Foundation of China under Grant No. 62502411, 2025 Guangdong Undergraduate University Teaching Quality and Teaching Reform Project, AI Research and Learning Base of Urban Culture under Project 2023WZJD008, Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007), and the Project of DEGP (No. 2023KCXTD042).

References

[1] Karan Ahuja, Deval Shah, Sujeath Pareddy, Franceska Xhakaj, Amy Ogan, Yuvraj Agarwal, and Chris Harrison. 2021. Classroom Digital Twins with Instrumentation-Free Gaze Tracking. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 484, 9 pages. doi:10.1145/3411764.3445711

[2] Moayad Aloqaily, Ouns Bouachir, Fakhri Karray, Ismaeel Al Ridhawi, and Abdulmotaleb El Saddik. 2022. Integrating digital twin and advanced intelligent technologies to realize the metaverse. *IEEE Consumer Electronics Magazine* 12, 6 (2022), 47–55.

[3] Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, et al. 2025. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558* (2025).

[4] Ayush Bhardwaj, Ashish Pratap, Edilberto F. Carrizales, Dongbeom Ko, Sungjoo Kang, and Jin Ryong Kim. 2025. MetaTwin: A Collaborative XR Platform for Seamless Physical-Virtual Synchronization. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 3, Article 70 (Sept. 2025), 32 pages. doi:10.1145/3749533

[5] Frank Biocca. 1997. The cyborg's dilemma: Progressive embodiment in virtual environments. *Journal of computer-mediated communication* 3, 2 (1997), JCMC324.

[6] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. 2025. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 16240–16250.

[7] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*. PMLR, 287–318.

[8] Zhejia Cai, Puhua Jiang, Shiwei Mao, Hongkun Cao, and Ruqi Huang. 2025. Improving Multi-View Reconstruction via Texture-Guided Gaussian-Mesh Joint Optimization. arXiv:2511.03950 [cs.CV] https://arxiv.org/abs/2511.03950

[9] Ziang Cao, Zhaoxi Chen, Liang Pan, and Ziwei Liu. 2025. PhysX-3D: Physical-Grounded 3D Asset Generation. arXiv:2507.12465 [cs.CV] https://arxiv.org/abs/2507.12465

[10] Ziang Cao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. 2025. PhysX-Anything: Simulation-Ready Physical 3D Assets from Single Image. arXiv:2511.13648 [cs.CV] https://arxiv.org/abs/2511.13648

[11] Stuartk Card, THOMASP MORAN, and Allen Newell. 1986. The model human processor- An engineering model of human performance. *Handbook of perception and human performance.* 2, 45–1 (1986), 1–35.

[12] Georgia Chalvatzaki, Ali Younes, Daljeet Nandha, An Thai Le, Leonardo FR Ribeiro, and Iryna Gurevych. 2023. Learning to reason over scene graphs: a case study of finetuning GPT-2 into a robot language model for grounded task planning. *Frontiers in Robotics and AI* 10 (2023), 1221739.

[13] Amanda Chan, Catherine Di, Joseph Rupertus, Gary D Smith, Varun Nagaraj Rao, Manoel Horta Ribeiro, and Andrés Monroy-Hernández. 2025. Redefining Research Crowdsourcing: Incorporating Human Feedback with LLM-Powered Digital Twins: Incorporating Human Feedback with LLM-Powered Digital Twins. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 454, 10 pages. doi:10.1145/3706599.3720269

[14] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. 2023. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 11509–11522.

[15] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14455–14465.

[16] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 18558–18568.

[17] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. 2022. TANGO: text-driven photorealistic and robust 3D stylization via lighting decomposition. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 2242, 14 pages.

[18] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. 2024. GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21476–21485.

[19] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. 2019. Holistic++ Scene Understanding: Single-View 3D Holistic Scene Parsing and Human Pose Estimation With Human-Object Interaction and Physical Commonsense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[20] Kevin Cheng and Paul A. Cairns. 2005. Behaviour, realism and immersion in games. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) *(CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1272–1275. doi:10.1145/1056808.1056894

[21] Deepayan Das, Davide Talon, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. 2025. One vlm to keep it learning: Generation and balancing for data-free continual visual question answering. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 5635–5645.

[22] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. LLMR: Real-time Prompting of Interactive Worlds using Large Language Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 600, 22 pages. doi:10.1145/3613904.3642579

[23] Mustafa Doga Dogan, Eric J Gonzalez, Karan Ahuja, Ruofei Du, Andrea Colaço, Johnny Lee, Mar Gonzalez-Franco, and David Kim. 2024. Augmented Object Intelligence with XR-Objects. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 19, 15 pages. doi:10.1145/3654777.3676379

[24] Xianzhe Dong, Tongxuan Liu, Yuting Zeng, Liangyu Liu, Yang Liu, Siyu Wu, Yu Wu, Hailong Yang, Ke Zhang, and Jing Li. 2025. HydraInfer: Hybrid Disaggregated Scheduling for Multimodal Large Language Model Serving. arXiv:2505.12658 [cs.DC] https://arxiv.org/abs/2505.12658

[25] Florian Dufresne, Charlotte Dubosc, Geoffrey Gorisse, and Olivier Christmann. 2024. Understanding the Impact of Coherence between Virtual Representations and Possible Interactions on Embodiment in VR: an Affordance Perspective. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 353, 7 pages. doi:10.1145/3613905.3650752

[26] Abdelrahman Elskhawy, Mengze Li, Nassir Navab, and Benjamin Busam. 2025. PRISM-0: A Predicate-Rich Scene Graph Generation Framework for Zero-Shot Open-Vocabulary Tasks. *arXiv preprint arXiv:2504.00844* (2025).

[27] Carmine Elvezio, Mengu Sukan, Ohan Oda, Steven Feiner, and Barbara Tversky. 2017. Remote collaboration in AR and VR using virtual replicas. In *ACM SIGGRAPH 2017 VR Village* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. doi:10.1145/3089269.3089281

[28] Shaojing Fan, Tian-Tsong Ng, Bryan Lee Koenig, Jonathan Samuel Herberg, Ming Jiang, Zhiqi Shen, and Qi Zhao. 2018. Image Visual Realism: From Human Perception to Machine Computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 9 (2018), 2180–2193. doi:10.1109/TPAMI.2017.2747150

[29] Shaojing Fan, Rangding Wang, Tian-Tsong Ng, Cheston Y.-C. Tan, Jonathan S. Herberg, and Bryan L. Koenig. 2014. Human Perception of Visual Realism for Photo and Computer-Generated Face Images. *ACM Trans. Appl. Percept.* 11, 2, Article 7 (July 2014), 21 pages. doi:10.1145/2620030

[30] Cathy Mengying Fang, Patrick Chwalek, Quincy Kuang, and Pattie Maes. 2024. WatchThis: A Wearable Point-and-Ask Interface powered by Vision-Language Models for Contextual Queries *(UIST Adjunct '24)*. Association for Computing Machinery, New York, NY, USA, Article 54, 4 pages. doi:10.1145/3672539.3686776

[31] Shuangkang Fang, I Shen, Yufeng Wang, Yi-Hsuan Tsai, Yi Yang, Shuchang Zhou, Wenrui Ding, Takeo Igarashi, Ming-Hsuan Yang, et al. 2025. MeshLLM: Empowering Large Language Models to Progressively Understand and Generate 3D Mesh. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14061–14072.

[32] Ylva Ferstl, Sean Thomas, Cédric Guiard, Cathy Ennis, and Rachel McDonnell. 2021. Human or Robot? Investigating voice, appearance and gesture motion realism of conversational social agents. In *Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents* (Virtual Event, Japan) *(IVA '21)*. Association for Computing Machinery, New York, NY, USA, 76–83. doi:10.1145/3472306.3478338

[33] Maia Garau, Mel Slater, Vinoba Vinayagamoorthy, Andrea Brogni, Anthony Steed, and M. Angela Sasse. 2003. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 529–536. doi:10.1145/642611.642703

[34] James J Gibson. 1979. The ecological approach to visual perception. (1979).

[35] James J Gibson. 2014. The theory of affordances:(1979). In *The people, place, and space reader*. Routledge, 56–60.

[36] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5021–5028.

[37] Junfu Guo, Yu Xin, Gaoyi Liu, Kai Xu, Ligang Liu, and Ruizhen Hu. 2025. ArticulatedGS: Self-supervised Digital Twin Modeling of Articulated Objects using 3D Gaussian Splatting. *arXiv preprint arXiv:2503.08135* (2025).

[38] Wen Hai, Nisha Jain, Andrzej Wydra, Nadia Magnenat Thalmann, and Daniel Thalmann. 2018. Increasing the feeling of social presence by incorporating realistic interactions in multi-party VR. In *Proceedings of the 31st International Conference on Computer Animation and Social Agents*. 7–10.

[39] Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Hui Liu, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and Jiliang Tang. 2025. Reasoning with Graphs: Structuring Implicit Knowledge to Enhance LLMs Reasoning. arXiv:2501.07845 [cs.CL] https://arxiv.org/abs/2501.07845

[40] Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. 2025. RoboGround: Robotic Manipulation with Grounded Vision-Language Priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 22540–22550.

[41] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, et al. 2023. Grounded decoding: Guiding text generation with grounded models for embodied agents. *Advances in Neural Information Processing Systems* 36 (2023), 59636–59661.

[42] Xincheng Huang, Michael Yin, Ziyi Xia, and Robert Xiao. 2024. VirtualNexus: Enhancing 360-Degree Video AR/VR Collaboration with Environment Cutouts and Virtual Replicas. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 55, 12 pages. doi:10.1145/3654777.3676377

[43] Robert JK Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 201–210.

[44] Senthil Kumar Jagatheesaperumal, Zhaohui Yang, Qianqian Yang, Chongwen Huang, Wei Xu, Mohammad Shikh-Bahaei, and Zhaoyang Zhang. 2023. Semantic-aware digital twin for metaverse: A comprehensive review. *IEEE Wireless Communications* 30, 4 (2023), 38–46.

[45] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. 2023. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241* (2023).

[46] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. 2025. PhysTwin: Physics-Informed Reconstruction and Simulation of Deformable Objects from Videos. *ICCV* (2025).

[47] Haiyan Jiang, Dongdong Weng, Xiaonuo Dongye, Le Luo, and Zhenliang Zhang. 2023. Commonsense Knowledge-Driven Joint Reasoning Approach for Object Retrieval in Virtual Reality. *ACM Trans. Graph.* 42, 6, Article 198 (Dec. 2023),

18 pages. doi:10.1145/3618320

[48] Haiyan Jiang, Dongdong Weng, Xiaonuo Dongye, Nan Zhang, and Luo Le. 2023. A commonsense knowledge-based object retrieval approach for Virtual reality. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 795–796.

[49] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. 2024. VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 78, 1 pages. doi:10.1145/3641519.3657448

[50] Crescent Jicol, Christopher Clarke, Emilia Tor, Rebecca M Dakin, Tom Charlie Lancaster, Sze Tung Chang, Karin Petrini, Eamonn O'Neill, Michael J Proulx, and Christof Lutteroth. 2023. Realism and Field of View Affect Presence in VR but Not the Way You Think. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 399, 17 pages. doi:10.1145/3544548.3581448

[51] Philip N Johnson-Laird. 2001. Mental models and deduction. *Trends in cognitive sciences* 5, 10 (2001), 434–442.

[52] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. 2023. Open-Vocabulary Object Detection upon Frozen Vision and Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=MIMwy4kh9lf

[53] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. 2025. STream3R: Scalable Sequential 3D Reconstruction with Causal Transformer. arXiv:2508.10893 [cs.CV] https://arxiv.org/abs/2508.10893

[54] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. 2025. Articulate-Anything: Automatic Modeling of Articulated Objects via a Vision-Language Foundation Model. arXiv:2410.13882 [cs.CV] https://arxiv.org/abs/2410.13882

[55] Guanghao Li, Kerui Ren, Linning Xu, Zhewen Zheng, Changjian Jiang, Xin Gao, Bo Dai, Jian Pu, Mulin Yu, and Jiangmiao Pang. 2025. ARTDECO: Towards Efficient and High-Fidelity On-the-Fly 3D Reconstruction with Structured Scene Representation. arXiv:2510.08551 [cs.CV] https://arxiv.org/abs/2510.08551

[56] Wenhao Li, Zhiyuan Yu, Qijin She, Zhinan Yu, Yuqing Lan, Chenyang Zhu, Ruizhen Hu, and Kai Xu. 2024. LLM-enhanced Scene Graph Learning for Household Rearrangement. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 32, 11 pages. doi:10.1145/3680528.3687607

[57] Yuqi Li, Chuanguang Yang, Junhao Dong, Zhengtao Yao, Haoyan Xu, Zeyu Dong, Hansheng Zeng, Zhulin An, and Yingli Tian. 2025. AMMKD: Adaptive Multimodal Multi-teacher Distillation for Lightweight Vision-Language Models. arXiv:2509.00039 [cs.CV] https://arxiv.org/abs/2509.00039

[58] Yiming Li, Xiaoshan Yang, and Changsheng Xu. 2022. Dynamic Scene Graph Generation via Anticipatory Pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13864–13873. doi:10.1109/CVPR52688.2022.01350

[59] Zhe Li, Xiang Bai, Jieyu Zhang, Zhuangzhe Wu, Che Xu, Ying Li, Chengkai Hou, and Shanghang Zhang. 2025. URDF-Anything: Constructing Articulated Objects with 3D Multimodal Language Model. arXiv:2511.00940 [cs.RO] https://arxiv.org/abs/2511.00940

[60] Zisu Li, Jiawei Li, Zeyu Xiong, Shumeng Zhang, Faraz Faruqi, Stefanie Mueller, Chen Liang, Xiaojuan Ma, and Mingming Fan. 2025. InteRecon: Towards Reconstructing Interactivity of Personal Memorable Items in Mixed Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 833, 19 pages. doi:10.1145/3706598.3713882

[61] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. 2023. PARIS: Part-level Reconstruction and Motion Analysis for Articulated Objects. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[62] Yufei Liu, Junwei Zhu, Junshu Tang, Shijie Zhang, Jiangning Zhang, Weijian Cao, Chengjie Wang, Yunsheng Wu, and Dongjin Huang. 2024. Texdreamer: Towards zero-shot high-fidelity 3d human texture generation. In *European Conference on Computer Vision*. Springer, 184–202.

[63] Chuofan Ma, Yi Jiang, Xin Wen, Zehuan Yuan, and Xiaojuan Qi. 2023. Codet: Co-occurrence guided region-word alignment for open-vocabulary object detection. *Advances in neural information processing systems* 36 (2023), 71078–71094.

[64] Marius Matulis and Carlo Harvey. 2021. A robot arm digital twin utilising reinforcement learning. *Computers & Graphics* 95 (2021), 106–114.

[65] Leo McElroy and Lingdong Huang. 2023. PotScript: a visual grammar for sculpting with functions. In *Proceedings of the 8th ACM Symposium on Computational Fabrication*. 1–9.

[66] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2Mesh: Text-Driven Neural Stylization for Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13492–13502.

[67] Masahiro Mori, Karl F MacDorman, and Norri Kageki. 2012. The uncanny valley [from the field]. *IEEE Robotics & automation magazine* 19, 2 (2012), 98–100.

[68] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.

[69] Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. 2015. Virtual Replicas for Remote Assistance in Virtual and Augmented Reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 405–415. doi:10.1145/2807442.2807497

[70] Nami Ogawa, Takuji Narumi, Hideaki Kuzuoka, and Michitaka Hirose. 2020. Do You Feel Like Passing Through Walls?: Effect of Self-Avatar Appearance on Facilitating Realistic Behavior in Virtual Environments. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3313831.3376562

[71] Leif Oppermann, Florian Buchholz, and Yücel Uzun. 2023. Industrial Metaverse: Supporting remote maintenance with avatars and digital twins in collaborative XR environments. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 178, 5 pages. doi:10.1145/3544549.3585835

[72] Stephen E Palmer. 1999. *Vision science: Photons to phenomenology*. MIT press.

[73] Weikun Peng, Jun Lv, Cewu Lu, and Manolis Savva. 2025. Generalizable Articulated Object Reconstruction from Casually Captured RGBD Videos. *arXiv preprint arXiv:2506.08334* (2025).

[74] Shengyi Qian, Weifeng Chen, Min Bai, Xiong Zhou, Zhuowen Tu, and Li Erran Li. 2024. AffordanceLLM: Grounding Affordance from Vision Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 7587–7597.

[75] Luca Randazzo, Inaki Iturrate, Serafeim Perdikis, and J d R Millán. 2017. mano: A wearable hand exoskeleton for activities of daily living and neurorehabilitation. *IEEE Robotics and Automation Letters* 3, 1 (2017), 500–507.

[76] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. 2024. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. arXiv:2401.14159 [cs.CV] https://arxiv.org/abs/2401.14159

[77] Oindrila Saha, Grant Van Horn, and Subhransu Maji. 2024. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 17542–17552.

[78] Daniel M Shafer, Corey P Carbonara, and Michael F Korpi. 2019. Factors affecting enjoyment of virtual reality games: a comparison involving consumer-grade virtual reality technology. *Games for health journal* 8, 1 (2019), 15–23.

[79] Thomas B Sheridan et al. 1992. Musings on telepresence and virtual presence. *Presence Teleoperators Virtual Environ.* 1, 1 (1992), 120–125.

[80] Mel Slater, Sylvia Wilbur, et al. 1997. A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments* 6, 6 (1997), 603–616.

[81] Carolin Stellmacher, Feri Irsanto Pujianto, Tanja Kojic, Jan-Niklas Voigt-Antons, and Johannes Schöning. 2024. Experiencing Dynamic Weight Changes in Virtual Reality Through Pseudo-Haptics and Vibrotactile Feedback. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 421, 13 pages. doi:10.1145/3613904.3642552

[82] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 161–174. doi:10.1145/3332165.3347872

[83] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. 2024. LLaMA-Mesh: Unifying 3D Mesh Generation with Language Models. arXiv:2411.09595 [cs.LG] https://arxiv.org/abs/2411.09595

[84] Jing Wei, Sungdong Kim, Hyunhoon Jung, and Young-Ho Kim. 2024. Leveraging large language models to power chatbots for collecting user self-reported data. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW1 (2024), 1–35.

[85] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. 2023. OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[86] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2025. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 21469–21480.

[87] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition. 4389–4398.

[88] Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2024. VLM-Grounder: A VLM Agent for Zero-Shot 3D Visual Grounding. In *CoRL*.

[89] Ran Xu, Yan Shen, Xiaoqi Li, Ruihai Wu, and Hao Dong. 2024. Naturalvlm: Leveraging fine-grained natural language for affordance-guided visual manipulation. *IEEE Robotics and Automation Letters* (2024).

[90] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. 2020. RigNet: Neural Rigging for Articulated Characters. *ACM Trans. on Graphics* 39 (2020).

[91] Kosei Yamao, Daiju Kanaoka, Kosei Isomoto, Akinobu Mizutani, Yuichiro Tanaka, and Hakaru Tamukoh. 2024. Development of a saycan-based task planning system capable of handling abstract nouns. In *Proceedings of International Conference on Artificial Life & Robotics (ICAROB2024)*. ALife Robotics, OS15–4.

[92] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. 2024. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*. Springer, 162–179.

[93] Nick Yee, Jeremy N Bailenson, and Kathryn Rickertsen. 2007. A meta-analysis of the impact of the inclusion and realism of human-like faces on user experiences in interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/1240624.1240626

[94] Andre Zenner and Antonio Krüger. 2017. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE transactions on visualization and computer graphics* 23, 4 (2017), 1285–1294.

[95] Michael JQ Zhang and Eunsol Choi. 2025. Clarify When Necessary: Resolving Ambiguity Through Interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 5526–5543. doi:10.18653/v1/2025.findings-naacl.306

[96] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. 2024. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*. Springer, 388–406.

[97] Xiaoyu Zhang, Fei Xue, Alexander Albers, and Torbjörn Netland. 2025. "It's impressive, but in practice...": Experiencing a Realistic Digital Transformation in and beyond the Classroom. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 14, 14 pages. doi:10.1145/3706598.3714169

[98] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, et al. 2024. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–18.

[99] Gengze Zhou, Yicong Hong, and Qi Wu. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 7641–7649.

[100] Zhengzhe Zhu, Ziyi Liu, Tianyi Wang, Youyou Zhang, Xun Qian, Pashin Farsak Raja, Ana Villanueva, and Karthik Ramani. 2022. MechARspace: An Authoring System Enabling Bidirectional Binding of Augmented Reality with Toys in Real-time. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 50, 16 pages. doi:10.1145/3526113.3545668

A Appendix

A.1 Questionnaire Questions and Semi-Structured Interview Questions

The goal of our user study is to evaluate the practical applicability of the framework and to explore a wider range of usage scenarios. Accordingly, we prioritize dimensions such as ease of use, understandability, and the system's ability to support users in realizing their intended outcomes. For this reason, we designed task-specific questions that remain grounded in core System Usability Scale (SUS) constructs while better capturing the unique experience of interacting with RealTwin. Moreover, RealTwin is intentionally designed as an AI-driven pipeline in which most reconstruction decisions are automated, and human involvement is limited to lightweight, corrective interactions. Under this interaction mode, SUS questions about unnecessary system complexity, dependence on technical support, or cumbersome manual operation are not relevant dimensions, because users are neither expected to operate the system extensively nor required to learn a complex interface. These constructs, therefore, are not included in our questionnaire.

Rate these statements (All 1 to 7 Likert scale from Strongly disagree to Strongly agree):

- **Q1:** "I feel easy when I use the framework." Corresponds to SUS constructs related to ease of use and learnability.
- **Q2:** "I am satisfied with the outcome of this framework." Relates to SUS constructs concerning integration of functions and user confidence.
- **Q3:** "I was able to improve the output iteratively." This item assesses iterative controllability, a core aspect of RealTwin's pipeline in which most decisions are automated by AI and human involvement consists of lightweight, corrective interventions.
- **Q4:** "I was able to achieve what I had in mind with this framework." This item evaluates the system's ability to support users in achieving their intended reconstruction goals.
- **Q5:** "The reconstruction process was intuitive."Aligns with SUS constructs regarding ease of use, learnability, and system consistency, capturing intuitiveness and process clarity.
- **Q6:** "I will recommend this tool to others." This item measures users' willingness to recommend the tool, reflecting perceived usefulness and overall acceptance, and encourages participants to consider broader usage scenarios for RealTwin.
- **Q7:** "I will use this tool in the future." Corresponds broadly to the intention-to-use dimension in the SUS, assessing long-term acceptance.

Semi-Structured Interview Questions:

- Could you walk me through how you used the framework? What steps did you take in the process?
- What kinds of objects would you most like to reconstruct with this framework, and why?
- In your view, does the RPDT created with this framework feel like a realistic digital twin? Why or why not?
- Have you had any previous experiences with digital twin creation or 3D reconstruction? What kinds of software or tools did you use, and how would you compare those experiences with using RealTwin?
- While working with the framework, were there any challenges you faced, or things you felt were difficult or not possible to achieve? How do you think it could be improved?
- Were there any parts of the framework that surprised you—something unexpected, either positive or negative?
- Thinking about your own research or work, in what contexts do you see this framework being useful or applied?
- Compared with traditional black-box AI models, what do you see as the strengths and weaknesses of a framework that allows human intervention in the process?

A.2 Failure Cases

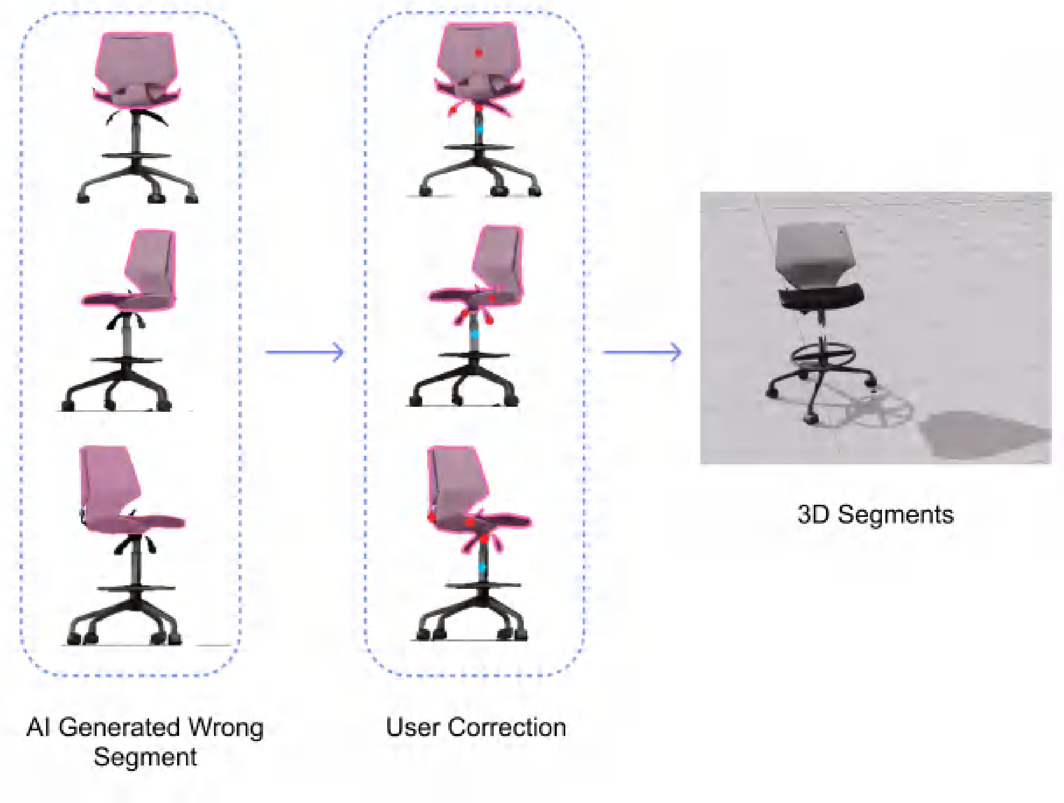


Figure 6: Mesh segmentation failure case. Users could refine the AI segments by interactively adding positive red points (regions expected to be inside the mask) and negative blue points (regions expected to be outside the mask) on the image.

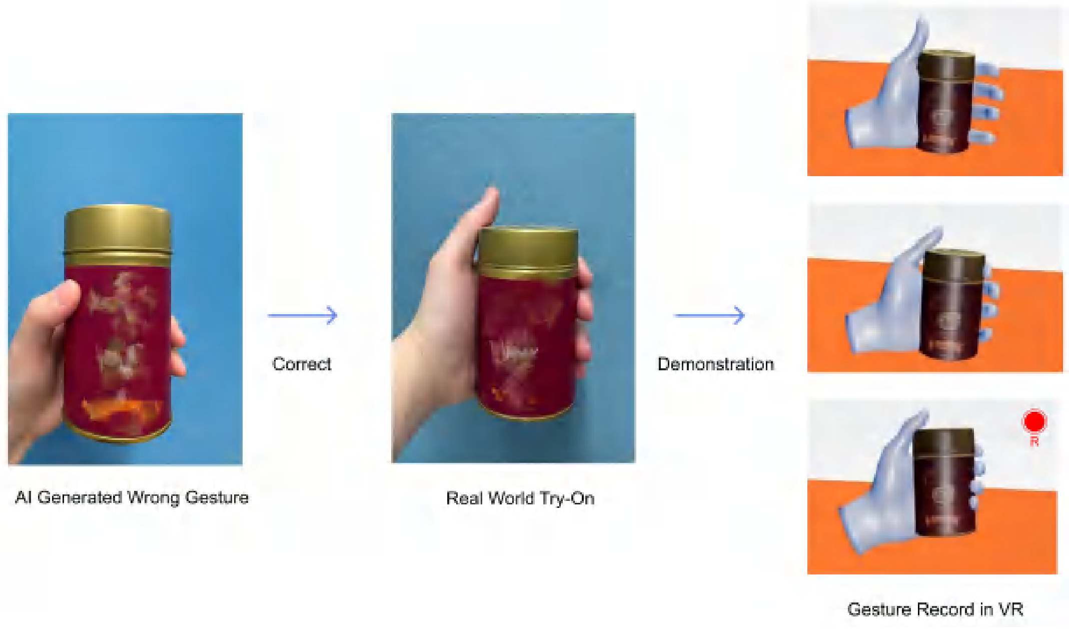


Figure 7: Hand affordance failure case. Users could directly demonstrate the intended gesture in the AR environment to bind the hand affordance to the target object.