DRAWER: Digital Reconstruction and Articulation With Environment Realism

Anonymous CVPR submission

Paper ID 4877

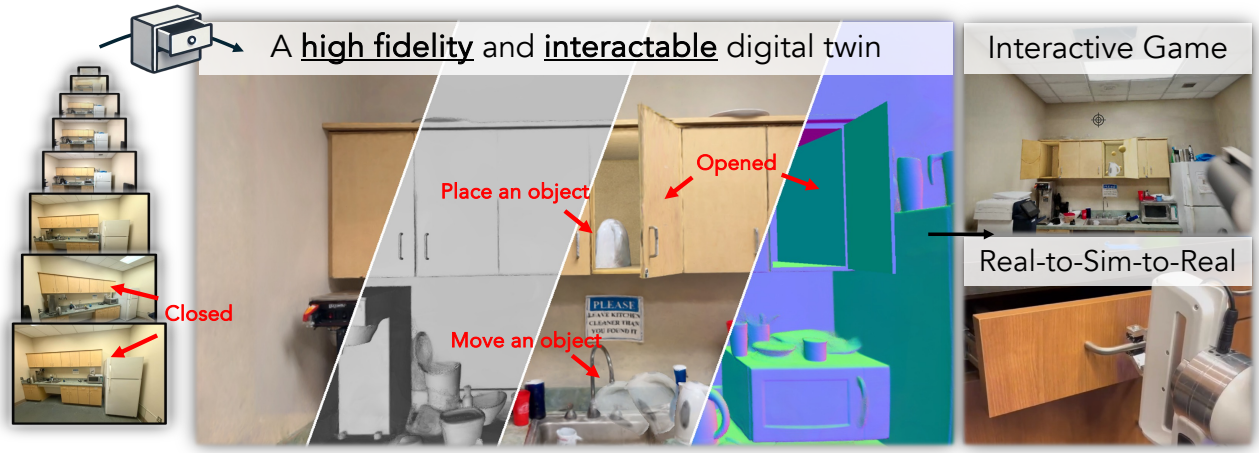


Figure 1. **DRAWER** automatically converts a video of a static scene into an interactive environment with segmented objects and articulated doors. It supports physical interactions like opening/closing drawers and moving/placing objects, with precise geometry and high-fidelity rendering. With these capabilities, DRAWER can transform a video into interactive games and enable real-to-sim-to-real transfer in robotics.

Abstract

Creating virtual digital replicas from real-world data unlocks significant potential across domains like gaming and robotics. In this paper, we present DRAWER, a novel framework that converts a video of a static indoor scene into a photorealistic *and* interactive *digital environment. Our approach centers on two main contributions: (i) a reconstruction module based on a* dual scene representation *that reconstructs the scene with* fine-grained geometric details, *and (ii) an* articulation *module that identifies articulation types and hinge positions, reconstructs simulatable shapes and appearances and integrates them into the scene. The resulting virtual environment is photorealistic, interactive, and runs in real time, with compatibility for game engines and robotic simulation platforms. We demonstrate the potential of DRAWER by using it to automatically create an interactive game in Unreal Engine and to enable real-to-sim-to-real transfer for robotics applications.*

1. Introduction

The ability to *automatically* create a *realistic*, *interactable*, and *highly detailed* virtual replica of a physical environment offers immense potential across multiple domains. For game developers, this presents an opportunity to replace painstaking human labor with streamlined, automated processes [14, 16]. In robotics, training and evaluating autonomous systems within richly detailed virtual spaces enable safer and more scalable learning. Take Fig. 1 as an example. Given a video of a scene (in this case, static), if we were to construct a digital replica that is not only visually and geometrically authentic but also physically grounded, then an agent deployed in this mirror world would be able to freely navigate the environment, interact with the scene (*e.g.*, opening drawers/cabinets, grabbing objects), and leverage observations and feedback to learn a policy that can seamlessly transfer to its real-world counterpart. Digital twins can thus serve as dynamic, virtual testbeds for studying and interacting with reality. However, to this day, automatically generating digital twins that mirror their real-world counterparts in terms of *visual appearance*, *geometric details*, and *physical properties* still remains a complex and unresolved task.

To mitigate the domain gap in visual or geometric quality, 3D reconstruction techniques, such as neural fields [53, 79], have emerged as promising solutions for constructing digital twins. However, despite their impressive realism, the reconstructions are still *static* and *non-actionable*. While users can

CVPR #4877

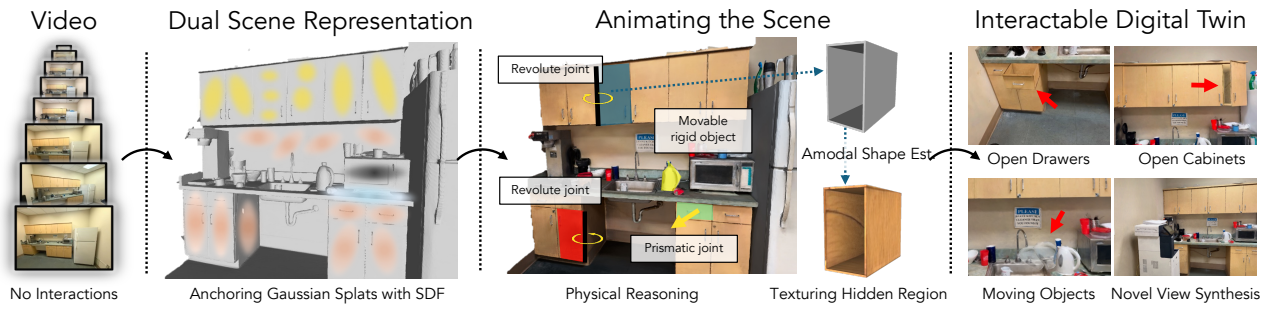


Figure 2. **Overview of DRAWER:** Given multiple posed images from a single video, we employ a *dual scene representation* that combines high-fidelity rendering with aligned geometry. We then *animate the scene* with physical reasoning to estimate articulated and movable rigid-body objects. Our amodel shape estimation with hidden region texturing enables us to create an *interactable digital twin*, supporting real-time physical interactions such as opening drawers/cabinets, moving objects, and rendering novel views.

freely view the scene from different angles, they cannot *interact* with it. Furthermore, there is often a trade-off between visual fidelity and geometric precision: pushing for photorealism often requires sacrificing some underlying geometric accuracy [31, 53], and vice versa [81, 97]. On the other hand, to enable interaction capabilities in digital representations, researchers have utilized shape primitives and CAD models to approximate the physical world [10, 11]. While they can construct virtual scenes and objects that resemble their real-world counterparts semantically and functionally, it comes at the expense of visual and geometric fidelity. The status quo calls for a method that takes the best of both worlds.

Towards this goal, we present, DRAWER, a novel framework that automatically converts a video of a static indoor scene into a *photorealistic* and *interactable* digital environment with *fine-grained geometric details*. At the core of our approach lies two key components: (i) a 3D reconstruction module based on a *dual scene representation* and (ii) an *articulation* module. Given an input video, we first construct a neural signed distance field (SDF) that effectively captures the scene's geometric details. We then initialize and anchor Gaussian splats with the estimated surface. This allows us to avoid floaters and preserve well-behaved geometry while enjoying rendering quality and speed. To make the reconstructed scene interactable, we leverage foundation models to infer articulation types and hinges of objects in the scene and approximate them with shape primitives. To ensure seamless integration of articulated objects into the original reconstruction, we further exploit differentiable rendering to align both their geometry and appearance. To enhance realism, we also infer the complete geometry and appearance of hidden interior regions. The resulting environment is photorealistic, interactable, and runs in real time.

We evaluate the fidelity of our reconstructed digital twins based on visual realism, articulation accuracy, and the precision of simulated motions across six distinct kitchen environments. DRAWER significantly outperforms prior art across all metrics. To further validate the effectiveness of the generated simulation environments, we employ them to create an interactive game and to train robotic controllers using a real-to-sim-to-real loop. Experimental results indicate that DRAWER eliminates the need for tedious manual effort and hand-specification in both applications.

2. Related Works

Novel view synthesis (NVS): The ability to render a scene from new viewpoints using a set of pre-captured images [7, 19, 24, 36, 71] is essential for building digital twins. The core of NVS is the co-design of scene representations and rendering methods. Among common representations such as neural radiance fields [4, 5, 53, 55, 78], neural textured mesh [73, 84], geometry primitives [1, 9, 40, 58, 91, 107], and neural surface fields [12, 38, 60, 79, 81, 83, 93, 94, 96], Gaussian splatting [31] emerges as a promising choice, offering flexibility and real-time rendering. On the other hand, neural surface models [94] provide accurate and detailed geometry, making them well-suited for reconstruction, articulation, and physical simulation. In this paper, we introduce a novel dual representation that combines the best of both works. Our approach extends beyond standard NVS, enabling active simulation and counterfactual visualization as the scene is interacted with and modified.

Data-driven simulation: Learning-based simulation [2, 8, 46, 51, 69, 92] has become popular for its effectiveness in simulating dynamics [37, 45, 85], modeling lighting [41, 62], and generating outputs in response to counterfactual actions [44, 84, 92]. These methods have been applied in various domains, including content creation [26, 45], game development [84], robot learning [8, 51, 63, 88, 90, 92], and multimodal generation like LiDAR [42, 51, 86, 92, 109, 110]. Our work falls within this category. We enable realistic modeling, simulation, and rendering of articulated objects in static scenes, from and to photorealistic videos. To our knowledge, this is the first approach of its kind. The closest work to ours is Video2Game [84], which also aims to reconstruct an interactable 3D scene from a video of a static scene. However, there are three key differences: 1) we significantly enhance interactivity by simulating articulated objects; 2) we improve

visual quality with a novel dual representation; and 3) beyond real-time gaming, we show utility for robot learning, where robots practice opening drawers and cabinets in our simulated environment and transfer these skills to the real world in a zero-shot setting.

Articulation modeling and simulation: Creating interactable and articulated virtual scenes that resemble reality typically requires specialized skills, professional software, and extensive human efforts [14, 56, 74]. To address this challenge, researchers have proposed using automated tools, such as procedural generation [13, 67], or learning-based methods to directly model or approximate real-world environments [25, 29, 84, 92]. However, current research in robotics and vision primarily focuses on individual objects [10, 23, 28, 35, 44, 49, 59, 95] and is not directly applicable to larger scenes. While some methods focus on scene-level modeling, they often assume access to dynamic scenes before and after interactions [25, 29], or rely on human interventions [32, 75]. To scale this to the real world, creating simulatable scenes from passive observations starts to gain more attention [10, 11]. However, these methods focus on replicating real-world semantics and functionality, often neglecting visual and geometric fidelity. In contrast, our digital twins closely replicate real-world environments in visual, geometric, and physical detail, which is beneficial for fields where visual fidelity is essential, such as content creation and sim2real applications. Furthermore, as shown in our experiments, our approach, grounded in precise geometry, achieves superior accuracy in articulation reasoning compared to these other methods.

Controllable video generation: An alternative to modeling how our world works is to leverage video generative models to (implicitly) simulate various effects [20, 22, 34, 39, 80, 89, 104, 105, 108]. While existing methods produce promising image space dynamics, they lack access to internal states, which are essential for tasks like mobile manipulation. For instance, knowing if a robot has grasped an object or opened a drawer is critical. Additionally, generated frames often degrade in quality over longer time spans, and integrating video dynamics with physical models or simulation engines remains challenging. In contrast, our approach adheres to physical laws, is compatible with simulation engines, and provides access to underlying states beyond visual rendering.

3. DRAWER: Digital Reconstruction and Articulation With Environment Realism

Given a video of a static scene, our goal is to develop an *interactable* and *actionable* digital twin that replicates the 3D world *geometrically*, *photometrically*, *physically*, and *efficiently*. Based on the observation that existing approaches tend to either focus on appearance modeling while neglecting physical interaction [53], or prioritize interaction at the expense of realism [10], we carefully design our method to fulfill all essential properties needed for realistic, real-time interactive applications. At the core of our approach is a *compositional dual scene representation* that effectively and efficiently supports both sensor and physics simulations. By decomposing the world into individual entities and modeling them with diverse yet tightly coupled representations, we can capture various modalities (*e.g.*, RGB, depth) and enable physical interactions without compromising fidelity. Fig. 2 shows an overview of our approach.

3DOI [64] Ours

Figure 3. **Articulation Estimation:** We visualize the estimated revolute axes and articulated object masks produced by 3DOI [64] and DRAWER, demonstrating that DRAWER achieves more precise articulation estimation due to its underlying 3D geometry.

3.1. Preliminaries

Neural signed distance fields (SDFs): A neural SDF f_θ^{SDF} maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ and a view direction $\mathbf{d} \in \mathbb{R}^2$ to an RGB radiance $\mathbf{c} \in \mathbb{R}^3$ and a signed distance to the nearest surface $s \in \mathbb{R}$: $s, \mathbf{c} = f_\theta^{\text{SDF}}(\mathbf{x}, \mathbf{d})$. One popular paradigm to learn neural SDF from a set of posed images is through volume rendering [52, 54, 93]. By converting signed distances to volume densities [79, 94], one can alpha-composite the radiance of 3D points $\mathbf{c}_i$ along each camera ray $\mathbf{r}$ to obtain the estimated pixel color $\mathbf{c}(\mathbf{r}) = \sum_{i=1}^{N} w_i \mathbf{c_i}$, and then compare with the GT: $\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r}} \|\hat{\mathbf{c}}(\mathbf{r}) - \mathbf{c}(\mathbf{r})\|_2^2$. Here, $w_i = \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j)$ indicates blending weight, and α_i represents opacity. We refer the readers to [93, 94] on how to derive opacity from signed distances. In practice, although Neural SDFs are better at capturing geometry, their rendering quality often lags behind NeRF-based approaches [4, 54]. Also, volume rendering requires sampling many points per ray, making it time-consuming and unsuitable for high-FPS applications. To improve efficiency, one strategy is to convert Neural SDFs into meshes. While this significantly accelerates rendering, it compromises rendering quality [84].

Gaussian splatting: An alternative approach for maintaining high visual quality with efficiency is to represent the scene as a set of 3D Gaussians [31]. Each Gaussian is characterized by a set of parameters: mean $\boldsymbol{\mu}$, scale $\mathbf{S}$, rotation $\mathbf{R}$, opacity α, and color radiance $\mathbf{c}$ (encoded using spherical harmonics). The covariance is derived as $\boldsymbol{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T$. By rasterizing the 3D Gaussians and alpha-compositing them, we obtain the pixel color $\mathbf{c}(\mathbf{p}) = \sum_{i=1}^{N} w_i \mathbf{c}_i$. We optimize all parameters by minimizing the photometric error $\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{p}} \|\hat{\mathbf{c}}(\mathbf{p}) - \mathbf{c}(\mathbf{p})\|_2^2$. A key property of Gaussian splatting is its support for adaptive density control. By

Method	Representation	PSNR↑	SSIM↑	LPIPS↓	Interactive Compatibility			
					Real time	Rigid-body physics	Scene decomposition	Articulation
Nerfacto [72]	Volume	25.49	0.911	0.163	✗	✗	✗	✗
Video2Game (NeRF) [84]	Volume	27.95	0.884	0.239	✗	✗	✗	✗
BakedSDF* [94]	Mesh	21.11	0.787	0.409	✓	✓	✗	✗
Video2Game (Mesh) [84]	Mesh	22.63	0.822	0.323	✓	✓	✓	✗
3DGS [31]	Points	30.42	0.954	0.126	✓	✗	✗	✗
2DGS* [27]	Points	25.68	0.884	0.269	✓	✓	✗	✗
Ours	Points+Mesh	27.80	0.912	0.159	✓	✓	✓	✓

Table 1. **Quantitative Results on Novel View Synthesis and Interactive Compatibility Analysis:** The dual scene representation in DRAWER provides competitive high-fidelity rendering with most-capable interactive compatibility. *BakedSDF [94] and 2D-GS [27] represent the entire scene as a whole, limiting object-level interactions.

dynamically spawning new Gaussians and culling redundant ones, the method effectively synthesizes both low- and high-frequency details. Gaussian splats are also inherently composible and controllable [47, 85]. While Gaussian splatting enables real-time, photorealistic rendering of complex scenes, the underlying geometry can be unsatisfactory. In practice, there are often 'floating' Gaussians in free space misaligned with the underlying geometry [21, 31, 98].

3.2. Dual Scene Representation

Each 3D representation has its own advantages and limitations, often involving a trade-off among visual fidelity, geometric precision, and speed. To address these inherent constraints, we propose leveraging different representations to capture each individual aspect while enforcing tight coupling between them.

Geometry: Accurate surface modeling is essential for physical simulation, such as collision modeling and object manipulation. To capture fine-grained geometric details, we first follow Yariv *et al*. [94] and parameterize the scene with a neural SDF. Since learning purely from RGB often leads to ambiguities [84, 97], we further leverage off-the-shelf 2D foundation models [3, 18] to predict surface normal and depth as regularization. We volume render the scene's color, depth, and normal and learn the SDF by jointly optimizing all the losses. Please refer to the supp. material for details.

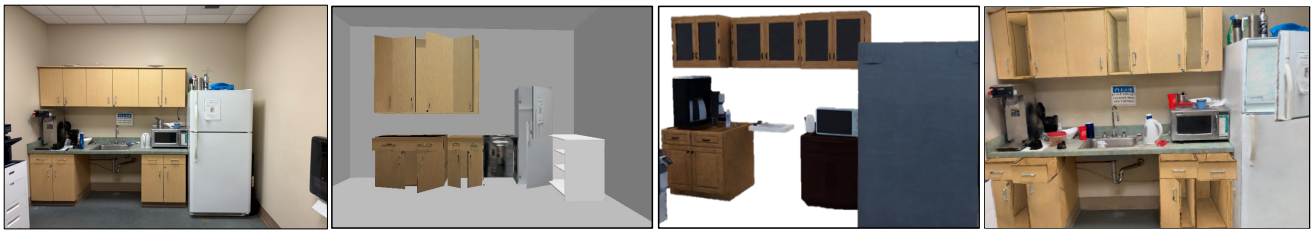
Appearance: High visual quality can significantly enhance immersive experiences in gaming and is essential for sim-to-real visual policy learning. We adopt 3D Gaussian splatting [31] due to its exceptional efficiency and ability to capture nuanced elements. Besides photometric error, we render Gaussian depth maps and use SDF depth rendering to regularize the Gaussians.

Coupling: One straightforward approach is to align the coordinates of Gaussian splatting with neural SDF, using the former for RGB rendering and the latter for collision modeling. However, this can lead to significant mismatch between visual observations and the actual geometry. For example, floating Gaussians may produce visual artifacts when viewed from different angles, creating the illusion of an object in free space [21, 26, 85]. This inconsistency is suboptimal for downstream applications.

To address this issue, we propose anchoring the Gaussians around the zero level-set of the neural SDF. This approach offers two main advantages: First, it allows the Gaussians to retain some flexibility in movement while avoiding the aforementioned issue. Second, if the scene is interacted with and the underlying scene SDF changes, anchoring the Gaussians to the SDF ensures that appearance changes are automatically handled. Since repeatedly querying the learned SDF is computationally expensive, we instead extract a high-resolution mesh from the SDF and anchor Gaussians to it. Specifically, we spawn Gaussians at the centroid of each face. The scales $\mathbf{S}$ are initialized to the respective face inradii, and the rotations $\mathbf{R}$ are aligned with the face normals. These Gaussians can move freely within the face and a limited distance along the normal direction. They can also tilt around the normal direction. To better align Gaussians with the underlying geometry, we regularize the scale along normal directions. For adaptive density control, during splitting, we ensure that new Gaussians remain on the same face and close to the existing one. We divide the scale by 1.6 and copy the rest of the remaining parameters. For more details, please refer to the supp. material.

Learning with straight-through estimator: Restricting each Gaussian to lie within a certain range of its corresponding face is a non-differentiable operation, which makes naive training of Gaussian splatting ineffective (see Sec. 4). To address this, we reparameterize all forward operations that involve $\texttt{clip}$ from $x^{\text{o}} = \texttt{clip}(x^{\text{in}})$ to $x^{\text{o}} = \texttt{sg}(\texttt{clip}(x^{\text{in}})) + x^{\text{in}} - \texttt{sg}(x^{\text{in}})$, where $\texttt{sg}(\cdot)$ denotes stop gradient. We then apply straight-through estimator [6] to transfer gradients from after clipping to before clipping.

Relationship to existing work: Our approach is closely related to recent work that combines 3D Gaussian splats with meshes [17, 26, 66, 82]. However, there exist several key differences. First, previous work either *fixes* the positions of Gaussians to the centroids of faces [61, 82] or employs a position loss to *encourage* Gaussians to remain close to the mesh. The former sacrifices flexibility, while the latter



Input URDFormer* [10] Digital Cousin [11] Ours

Figure 4. **Qualitative Results on Interactable 3D Reconstruction:** We compare DRAWER's reconstruction with [10, 11] which uses multiple images as input and selects the best output. DRAWER achieves realistic, well-aligned results, while [10, 11] show misalignment and lack realism. * We manually annotate bounding boxes for URDFormer [10] to improve their performance. To view this figure as a **video**, we recommend using Adobe Acrobat.

cannot guarantee effective coupling. In contrast, we allow Gaussians to move freely within the face and a limited distance along the normal direction, ensuring a balance between flexibility and geometric binding. Second, prior work subdivides meshes a priori to obtain denser Gaussian placements [17], whereas we use adaptive density control to automatically spawn new Gaussians as needed. As we will show in Sec. 4, these design differences have a significant impact on the final rendering quality.

3.3. Articulating the Scene

Now that we have a dual scene representation of a *static* scene, with high-quality visual appearance and detailed geometric structure, the next step is to estimate the underlying physical properties, such as articulation, and make the scene *actionable* and *interactable*.

Scene decomposition: The first step is to identify potential interactable objects and segment them in 3D space. In this work, we focus on both articulated objects (*e.g.*, drawers and cabinets) and rigid objects (*e.g.*, cups and bottles). Here, we primarily discuss the processing of articulated objects, while details on rigid objects, which follow a similar pipeline, are provided in the supp. material.

Given a set of posed images, we first adopt Grounded SAM [33, 43, 68] to segment all objects of interest across all frames. Due to viewpoint variations and specular reflections, objects are not always fully visible, resulting in significant variation in mask quality. To filter out unreliable estimations and associate masks across frames, we project all masks onto the 3D mesh (obtained in Sec. 3.2) and fuse them using the Louvain algorithm [76, 102]. We discard masks whose IoU with their fusion results fall below a threshold. Given the superior visual grounding capabilities of VLMs [87], we further employ GPT4o to assess mask quality and filter out unreliable ones. We then exploit SAM [33] to re-segment objects using point prompts derived from the fusion results. This process yields, for each object i, its high-quality 2D masks $\mathbf{M}_{i,j}$ in each image j, and its partial 3D geometry in either mesh form $\mathcal{M}_i^{\text{obj}} = (\mathbf{V}_i, \mathbf{F}_i)$ or SDF.

Physical reasoning: Once we identify all interactable objects, the next step is to estimate their physics-related attributes to effectively model and simulate their physical dynamics. We adopt a two-pronged approach to estimate articulation types and axes of articulated objects. The first prong leverages a specialized vision foundation model 3DOI [64] to predict object hinges and affordances. We integrate its predictions with the underlying 3D to improve accuracy. Since 3DOI's performance varies by viewpoint, we use GPT4o for a more robust second estimation. When results differ, another VLM arbitrates the final decision. This enhances overall accuracy. For other physical parameters (*e.g.*, mass, friction), following [84, 101], one can either obtain estimates through VLM queries or set them manually.

Amodal shape estimation: Having established object articulations, we now face the challenge of hidden regions. When interacting with a reconstructed scene, previously invisible regions in the input video may become exposed. For instance, when cabinet doors open, their interior surfaces become visible. Without proper modeling, the geometry and appearance of these originally hidden regions would be underconstrained. This limitation prevents more sophisticated interactions, such as picking up a mug from the countertop and placing it into a drawer.

To address this issue, we first define a compositional 3D template for each object category $\mathcal{M}^{\text{tmp}} = (\mathbf{V}^{\text{tmp}}, \mathbf{F}^{\text{tmp}}) = \{(\mathbf{V}_i^{\text{part}}, \mathbf{F}_i^{\text{part}})\}_{i=1}^{K}$. Then, we exploit VLMs to refine the structure (*e.g.*, adjusting the number of layers a cabinet has). Since the templates comprise well-defined shape primitives, we can easily edit the compositional structure or modify their shape to better match different observations.

For each object i, we consider three objectives: (i) the *mask consistency term* measures the discrepancy between the rendered masks and the observed masks: $\mathcal{L}_{\text{mask}} = \sum_j \|\mathbf{M}_{i,j} - \text{Rend}(\mathcal{M}_i^{\text{tmp}})\|_2^2$; (ii) the *shape consistency term* encourages the visible part of the template to match its corresponding partial 3D geometry in the scene. Since densely querying the learned SDF is computationally expensive, we instead adopt Chamfer Distance (CD): $\mathcal{L}_{\text{shape}} = \text{CD}(\mathcal{M}_i^{\text{tmp, vis}}, \mathcal{M}_i^{\text{obj}})$; and (iii) the *structure consistency term* encourages originally adjacent parts α, β to remain adjacent

Figure 5. **Qualitative Results on Articulation Simulation:** Our method achieves realistic, accurate articulation, while KlingAI fails despite using manual segmentation masks and motions.

Figure 6. **DRAWER in Unreal Engine.** We demonstrate our interactive game in Unreal Engine with game features including shooting rigid objects like the blue bottle and white kettle segmented from the scene as well as opening cabinet and drawer doors.

after optimization: $\mathcal{L}_{\text{struc}} = \sum_{(\alpha,\beta)} \sum_{(j,k)} \|\mathbf{V}^{\alpha}_{i,j} - \mathbf{V}^{\beta}_{i,k}\|_2^2$, where (j, k) are the vertex pairs from α, β that are within a certain distance threshold. We use PyTorch3D as the differentiable renderer. We optimize the poses and shape parameters (*e.g.*, scale, width, length, etc) of all parts. We start with $\mathcal{L}_{\text{shape}}$ and $\mathcal{L}_{\text{struc}}$, and then turn on all objectives. To ensure objects do not collide with each other, we further adopt a global regularization term to penalize inter-penetration [57].

Texturing: We exploit MatFuse [77], a conditional diffusion model, to estimate the PBR materials of unobserved regions. We can either parameterize them as texture maps or distill them back to Gaussians in our dual representation.

Composing back to scene: Directly editing and merging meshes is extremely difficult due to changes in topology. Fortunately, our dual representation, which builds on neural SDF and Gaussian splatting, is inherently compositional in 3D. Furthermore, meshes and SDFs are largely interchangeable. Therefore, we can convert the completed articulated object back to the dual representation and use it to replace the original partial one.

3.4. Downstream Applications with DRAWER

Gaming: Content creation that reflects real world diversity with visual and physical fidelity, is a challenging problem for gaming applications. We demonstrate the utility of environments created with DRAWER for gaming applications using Unreal Engine (UE) [15]. Specifically, we show how an agent dropped into the reconstructed environment imported into Unreal Engine can interact with the world to perform movement, shooting or opening of the various elements of the scene. Unreal engine offers support for rigid-body dynamics and articulation. Leveraging our dual representation—where Gaussians anchored to the SDF enable high-quality rendering, and SDF-derived mesh provides accurate collision geometry—we achieve alignment of rendering and physical models. The Luma Unreal Engine Plugin [48] allows real-time Gaussian rendering, while collision models use the SDF-extracted mesh. Articulation joints in UE are configured based on estimated types and axes, completing the interactive setup. As outlined in previous work [84], we can then develop an interactive agent that can navigate and interact with various elements of the scene.

Real-to-sim-to-real transfer for robot learning: Besides gaming applications, DRAWER holds value in data generation and model training for robotics. The environments created in DRAWER can be imported into a physics simulator such as Isaac Sim [56] with appropriate kinematics and dynamics. This enables the generation of physically realistic interaction data for tasks such as drawer opening/closing or object pick and place using motion planning or RL, without requiring tedious human effort. The generated data can be used to learn a policy that can be transferred to act in the real world directly from perception [29, 50, 75]. DRAWER generated environments offer greater visual and geometric fidelity than other environment creation methods, helping to bridge the simulation to reality gap. This circumvents much of the burden of real-world data collection on-robot.

4. Experiments

4.1. Setup

Dataset: We manually capture videos in six different kitchens. The scenes are static, with no assumed interactions, which significantly reduces the capture cost. For evaluation, we annotate the type of articulation and any hinges present on all articulated objects within each scene. Additionally, we use fishing wire to open cabinet doors and drawers, obtaining GT video snippets of object articulations. We further

Opening the drawer | Picking | Placing | Closing the drawer

Figure 7. **Scaling up robot learning data with our digital twin.** We demonstrate our interactive environment is capable of conducting robot learning tasks including opening/closing drawers and picking/placing segmented rigid objects in the scene. To view this figure as a **video**, we recommend using Adobe Acrobat.

Method	Representation	PSNR↑	SSIM↑	LPIPS↓
Textured mesh	Mesh	19.71	0.785	0.323
+ Static GS	Points+Mesh	25.87	0.889	0.276
+ Movable GS	Points+Mesh	26.49	0.896	0.193
+ Straight-through est.	Points+Mesh	**27.80**	**0.912**	**0.159**

Table 2. **Ablation Study on DRAWER Model Design:** We observe improved rendering quality with the sequential addition of each designed component in DRAWER.

annotate key points on moving surfaces (*e.g.*, cabinet doors, drawer fronts), fit homographies to these points, and extract dense 2D pixel trajectories. We use these trajectories to evaluate simulated articulated motions.

Metrics: We adopt PSNR, SSIM, and LPIPS [103] to assess the visual quality of digital twins. For articulation type estimation, we compute both precision and recall to explicitly account for potential perception errors (*e.g.*, missed segmentation), which frequently occur in real-world scenarios. To quantify the accuracy of articulated motion simulation in digital twins, we track pixels along moving surfaces (*e.g.*, cabinet doors) [30] and compute the Earth Mover's Distance (EMD) between simulated and ground truth trajectories. For revolute objects, we additionally report EA-Score [65, 106], which measures both the angular and Euclidean distance between predicted and ground truth rotation axes.

4.2. Experimental Results

Novel view synthesis: Tab. 1 compares the visual realism and interactive capabilities of our reconstructed twins with those of prior works. We evaluate state-of-the-art approaches that utilize different representations, including Nerfacto [72], BakedSDF [94], Video2Game [84], 3DGS [31], and 2D GS [27]. Although 3DGS achieves the best rendering results, it compromises geometric fidelity, leading to reconstructed scenes that are neither realistic nor interactive. In contrast, our approach strikes the balance between the two. We significantly outperform neural rendering methods that support high-quality geometry, while also providing the most interactive functionalities, beyond simple pick-and-place.

Articulation estimation: The performance of articulation estimation relies on both perception (*i.e.*, identifying articulated) and reasoning (*e.g.*, estimating articulation type). To validate the effectiveness of each module, we first assess the accuracy of our estimation *given the masks of objects of interest*. We compare with 3DOI [64], a foundation model for articulation prediction, in Tab. 4 and Fig. 3. Since our approach leverages an ensemble of multi-modal models, it is more robust and achieves higher accuracy. By grounding predictions in the underlying 3D, we further improve the precision of estimated rotation axes.

We then evaluate the full pipeline, comparing it with recent methods for interactable 3D reconstruction: URDFormer [10] and Digital Cousin [11]. As shown in Fig. 4 and Tab. 3, our approach not only accurately recalls most articulated objects with high precision but also maintains exceptionally high visual and geometric fidelity.

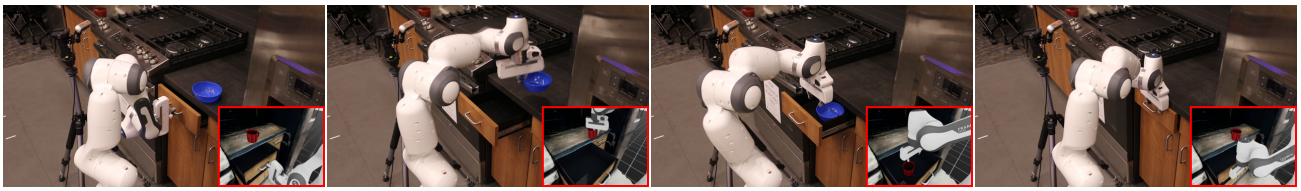
Articulated motion simulation: Besides articulation type, the quality of simulated articulation motions is also crucial for creating high-quality digital twins. Since the reconstructions in [10, 11] diverge significantly from the original scenes, they are not directly comparable. Instead, we compare our method with KlingAI[1], a SOTA conditional video diffusion model that supports motion control. We manually provide their model point prompts to segment objects of interest and specify the desired articulated motions. However, despite the privileged information, the synthesized motions are often infeasible. In contrast, our automatic approach is physicsally-grounded and outperforms KlingAI by an order of magnitude (EMD: 1.41×10^{-5} *v.s.* 17.7×10^{-5}). We show simulated motions in Fig. 5.

Ablation study: We start with a high-quality mesh extracted from a neural SDF and sequentially add back other components. As shown in Tab. 2, incorporating Gaussian splatting significantly improves overall performance. Additionally, allowing the Gaussians to move and using straight-through estimation further enhances the results.

4.3. Build Your Own Game

We have demonstrated our system's effectiveness in rendering quality and articulation inference accuracy across various setups. Next, we construct an interactive game with

[1] https://www.klingai.com/



Opening the drawer Picking Placing Closing the drawer

Figure 8. **Real-to-Sim-to-Real.** DRAWER allows us to learn train robotic controllers using a real-to-sim-to-real loop. The inset images indicate the simulated data generation process. To view this figure as a **video**, we recommend using Adobe Acrobat.

first-person player control and real-time interaction.

Data preparation: Our game assets are derived from self-captured kitchen videos (Sec. 4.1) and feature SDF-extracted mesh geometry for collision models, segmented objects for rigid-body dynamics, and articulated drawers with fully detailed interiors. Our dual-representation reconstruction enables high-quality Gaussian rendering in real time.

Interactive game features: As shown in Fig. 6, our game, built upon Unreal Engine (UE) (Sec. 3.4), supports high-quality rendering and diverse physical interactions at an interactive rate. Key features include: *Movement:* Players can navigate the room freely with realistic physics and collision models. *Shooting:* In a first-person view, players can shoot balls at segmented objects, with realistic motion simulated upon impact. *Opening:* Players can interact with articulated objects, such as drawers and cabinets, utilizing estimated articulations. Segmented items, like a kettle, can be removed from cabinets and dropped, with realistic dynamics and rendering enabled by our dual representation.

4.4. Real-to-Sim-to-Real

We conduct a proof of concept experiment with our articulated environment in a robotic real-to-sim-to-real setting. We reconstruct the scene with DRAWER, automatically generate simulation data via motion planning for policy learning, and transfer learned policies to the real world. A similar pipeline was shown in [75], with manual articulations.

Data generation: To generate the data for policy learning, we first import the geometry and articulation reconstructed from the real scene via the DRAWER into Isaac Sim [56]. As shown in Fig. 7, we then initialize the robot's pose around each drawer and utilize standard motion planning [70] in combination with a object-centric grasp sampler [99] to generate motion data. This approach allows for generating physically realistic data for tasks such as opening the drawer by pulling the handle, picking and placing objects inside the drawer, and closing the drawer autonomously, without requiring considerable manual human effort.

Policy learning: Given this data, we then train a policy to open and close the drawer using behavior cloning on the collected data. Rendering the generated scenes in simulation as a point cloud, we deploy a commonly used policy learning architecture based on 3D Diffusion Policies [100]. Mirroring the deployment setting, the policy takes in a cropped point cloud and proprioceptive information and predicts the next end-effector pose of the robot that is then executed on the robot via inverse kinematics and position control.

Real-world deployment: Finally, after training policies in simulation, we can deploy policies in a zero-shot fashion in the real world on a Franka Emika Panda robot mounted on a mobile base. In this instantiation, we trained independent policies for each substage of the problem - drawer closing, picking and placing, and opening. A qualitative visualization of the learned behavior is shown in Fig 8. We refer readers to supplementary material for a more detailed visualization.

Method	Total #	Pred #	Correct #	Precision (%)	Recall (%)
URDFormer* [10]	136	56	32	57.1	23.5
Digital Cousin* [11]		171	88	51.5	64.7
Ours		115	108	**93.9**	**79.4**

Table 3. **Articulation Understanding Comparison:** Total#, Pred#, and Correct# represent the total, predicted, and correctly predicted numbers of articulated objects. *We conduct multiple predictions for each scene (based on different views) and select the best result.

Method	Total #	Correct #↑	Rev. #	Correct Rev. #↑	EA-Score ↑
3DOI	80	**78**	59	57	0.861
Ours		**78**		**58**	**0.994**

Table 4. **Articulation Inference Comparison:** Total # and Correct # represent the total number of articulated objects evaluated and the number correctly predicted, while Rev. # denotes the count of correctly predicted revolute objects.

5. Conclusion

We present DRAWER, a novel framework that automatically converts a single video into an interactive environment with articulated and rigid-body dynamics, requiring no prior articulation data. Our method integrates an SDF field and Gaussian splats into a dual scene representation, which is then decomposed and articulated to create a functional environment. We demonstrate DRAWER's superior performance in articulation understanding and rendering, as well as its utility in developing realistic interactive games and enabling real-to-sim-to-real transfer for robot learning. Looking ahead, DRAWER could benefit from integration with more sophisticated relightable environment reconstruction.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics, 2020. 2

[2] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *ICRA*, 2022. 2

[3] Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, 2024. 4

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2, 3

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[6] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv*, 2013. 4

[7] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993. 2

[8] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *CVPR*, 2021. 2

[9] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *CVPR*, 2023. 2

[10] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv*, 2024. 2, 3, 5, 7, 8

[11] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Acdc: Automated creation of digital cousins for robust policy learning. *arXiv*, 2024. 2, 3, 5, 7, 8

[12] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022. 2

[13] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *NeurIPS*, 2022. 3

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. 1, 3

[15] Epic Games. Unreal engine. 6

[16] Rockstar Games. Grand theft auto v, 2014. 1

[17] Lin Gao, Jie Yang, Bo-Tao Zhang, Jia-Mu Sun, Yu-Jie Yuan, Hongbo Fu, and Yu-Kun Lai. Mesh-based gaussian splatting for real-time large-scale deformation. *arXiv*, 2024. 4, 5

[18] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. *arXiv*, 2024. 4

[19] Jonathan Shade Steven Gortler, Li-wei He, Richard Szeliski, et al. Layered depth images. In *SIGGRAPH*, pages 231–242, 1998. 2

[20] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023. 3

[21] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering, 2023. 4

[22] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018. 3

[23] David S Hayden, Jason Pacheco, and John W Fisher. Nonparametric object and parts modeling with lie group dynamics. In *CVPR*, 2020. 3

[24] Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and Luc Van Gool. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. In *DAGM-Symposium*, 1999. 2

[25] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *ICRA*, 2023. 3

[26] Hao-Yu Hsu, Zhi-Hao Lin, Albert Zhai, Hongchi Xia, and Shenlong Wang. Autovfx: Physically realistic video editing from natural language instructions, 2024. 2, 4

[27] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*. Association for Computing Machinery, 2024. 4, 7

[28] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *CVPR*, 2021. 3

[29] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *CVPR*, 2022. 3, 6

[30] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv*, 2024. 7

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 2, 3, 4, 7

[32] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024. 3

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 5

[34] KlingAI, 2024. 3

[35] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. *arXiv*, 2024. 3

[36] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[37] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Physically-based neural rendering for extreme climate synthesis. *arXiv*, 2022. 2

[38] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 2

[39] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dynamics. In *CVPR*, 2024. 3

[40] Zhi-Hao Lin, Wei-Chiu Ma, Hao-Yu Hsu, Yu-Chiang Frank Wang, and Shenlong Wang. Neurmips: Neural mixture of planar experts for view synthesis. In *CVPR*, 2022. 2

[41] Zhi-Hao Lin, Bohan Liu, Yi-Ting Chen, David Forsyth, Jia-Bin Huang, Anand Bhattad, and Shenlong Wang. Urbanir: Large-scale urban scene inverse rendering from a single video, 2023. 2

[42] Jeffrey Yunfan Liu, Yun Chen, Ze Yang, Jingkang Wang, Sivabalan Manivasagam, and Raquel Urtasun. Real-time neural rasterization for large scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8416–8427, 2023. 2

[43] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv*, 2023. 5

[44] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation, 2024. 2, 3

[45] Ziang Liu, Genggeng Zhou, Jeff He, Tobia Marcucci, Li Fei-Fei, Jiajun Wu, and Yunzhu Li. Model-based control with sparse neural dynamics, 2023. 2

[46] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives, 2023. 2

[47] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 4

[48] Luma AI. Luma unreal engine plugin. 6

[49] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. *arXiv*, 2024. 3

[50] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. *CoRR*, abs/2406.08474, 2024. 6

[51] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *CVPR*, 2020. 2

[52] Nelson Max. Optical models for direct volume rendering. *TOG*, 1995. 3

[53] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3

[54] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ACM Communications*, 2021. 3

[55] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022. 2

[56] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 2023. 3, 6, 8

[57] Lea Muller, Ahmed AA Osman, Siyu Tang, Chun-Hao P Huang, and Michael J Black. On self-contact and human pose. In *CVPR*, 2021. 6

[58] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 2

[59] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects. In *CVPR*, 2022. 3

[60] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[61] Pramish Paudel, Anubhav Khanal, Ajad Chhatkuli, Danda Pani Paudel, and Jyoti Tandukar. ihuman: Instant animatable digital humans from monocular videos. *arXiv*, 2024. 4

[62] Ava Pun, Gary Sun, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, Wei-Chiu Ma, and Raquel Urtasun. Lightsim: Neural lighting simulation for urban scenes, 2023. 2

[63] Ava Pun, Gary Sun, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, Wei-Chiu Ma, and Raquel Urtasun. Neural lighting simulation for urban scenes. *NeurIPS*, 2024. 2

[64] Shengyi Qian and David F Fouhey. Understanding 3d object interaction from a single image. In *ICCV*, 2023. 3, 5, 7

[65] Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F Fouhey. Understanding 3d object articulation in internet videos. In *CVPR*, 2022. 7

[66] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv*, 2024. 4

[67] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation, 2024. 3

[68] 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. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 5

[69] Sanghyun Son, Yi-Ling Qiao, Jason Sewall, and Ming C Lin. Differentiable hybrid traffic simulation. *TOG*, 2022. 2

[70] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023. 8

[71] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision*, 1998. 2

[72] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, et al. Nerfstudio: A modular framework for neural radiance field development. *in arXiv*, 2023. 4, 7

[73] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. Delicate textured mesh recovery from nerf via adaptive surface refinement. *arXiv preprint arXiv:2303.02091*, 2023. 2

[74] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012. 3

[75] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv*, 2024. 3, 6, 8

[76] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*. 5

[77] Giuseppe Vecchio, Renato Sortino, Simone Palazzo, and Concetto Spampinato. Matfuse: controllable material generation with diffusion models. In *CVPR*, 2024. 6

[78] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

[79] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv*, 2021. 1, 2, 3

[80] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2024. 3

[81] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction, 2023. 2

[82] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G. Schwing, and Shenlong Wang. Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh. *arXiv*, 2024. 4

[83] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. *arXiv preprint arXiv:2208.12697*, 2022. 2

[84] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time interactive realistic and browser-compatible environment from a single video. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7

[85] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics, 2024. 2, 4

[86] Yuwen Xiong, Jingkang Ma, Wei-Chiu Wang, and Raquel Urtasun. Ultralidar: Learning compact representations for lidar completion and generation. *CVPR*, 2023. 2

[87] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv*, 2023. 5

[88] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In *ICLR*, 2024. 2

[89] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv*, 2023. 3

[90] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. In *CVPR*, 2020. 2

[91] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving, 2020. 2

[92] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. *CVPR*, 2023. 2, 3

[93] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021. 2, 3

[94] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *SIGGRAPH Conference*, 2023. 2, 3, 4, 7

[95] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv*, 2018. 3

[96] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022. 2

[97] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv*, 2022. 2, 4

[98] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv*, 2024. 4

[99] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. In *7th Annual Conference on Robot Learning*. 8

[100] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024. 8

[101] Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *CVPR*, 2024. 5

[102] Jicun Zhang, Jiyou Fei, Xueping Song, and Jiawei Feng. An improved louvain algorithm for community detection. *Mathematical Problems in Engineering*, 2021. 5

[103] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[104] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-based interaction with 3d objects via video generation. In *ECCV*, 2024. 3

[105] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv*, 2023. 3

[106] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4793–4806, 2021. 7

[107] Yiqun Zhao, Chenming Wu, Binbin Huang, Yihao Zhi, Chen Zhao, Jingdong Wang, and Shenghua Gao. Surfel-based gaussian inverse rendering for fast and relightable dynamic human reconstruction from monocular video. *arXiv preprint arXiv:2407.15212*, 2024. 2

[108] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. 3

[109] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic lidar point clouds. In *ECCV*, 2022. 2

[110] Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. Lidardm: Generative lidar simulation in a generated world, 2024. 2