SIMFOUNDRY: Modular and Automated Scene Generation for Policy Learning and Evaluation

Nadun Ranawaka[1,2*], Josiah Wong[1,3*], Wei-Lin Pai[3], Wei-Teng Chu[3], Tianyuan Dai[1,4], Masoud Moghani[1,5], Hang Yin[3], Yunfan Jiang[1,3], Wesley Durbano[1*], Brandon Huynh[1*], Linxi Fan[1], Danfei Xu[1,2], Ruohan Zhang[3], Li Fei-Fei[3], Bowen Wen[1], Ajay Mandlekar[1†], Yuke Zhu[1,4†]

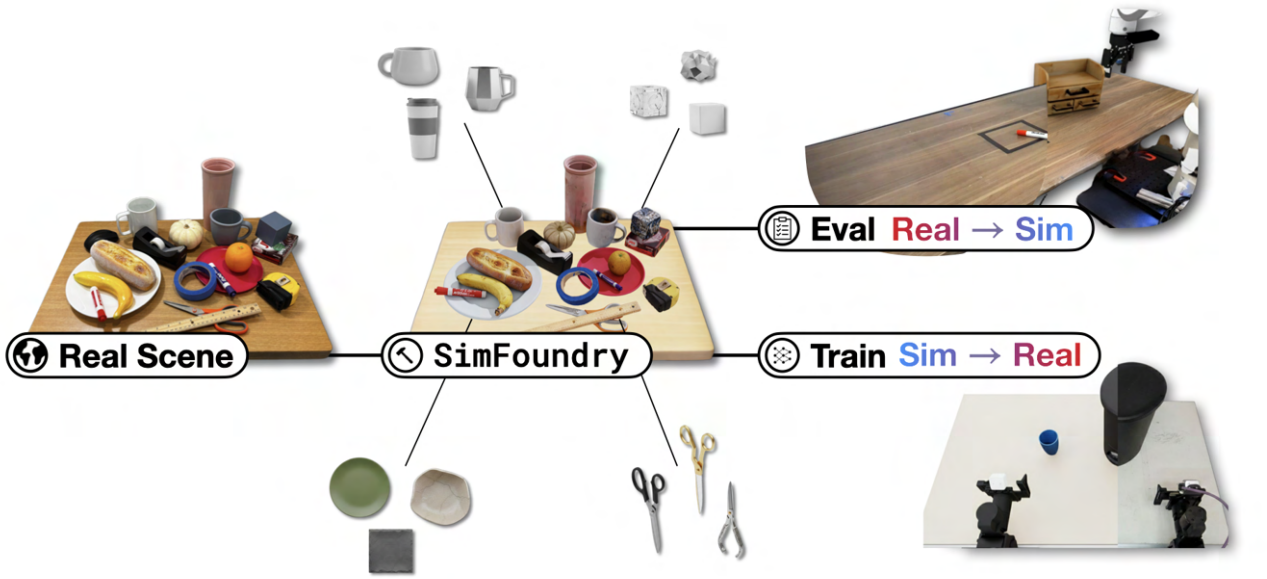


Fig. 1: **Overview.** SIMFOUNDRY generates an unlimited number of fully-interactive, sim-ready scenes from a single input video. These scenes can be applied to both evaluate real-world trained policies, as well as train simulation policies that transfer zero-shot to the real world.

***Abstract*— Training and evaluating robot policies in the real world can be costly and difficult to scale. Simulation offers a sandbox for efficient policy evaluation and generation of policy training samples, but traditionally requires substantial engineering effort to construct simulated environments that align with reality. While this burden has been alleviated by prior work proposing end-to-end methods for generating pre-processed and annotated "sim-ready" scenes useful for downstream robotics tasks, these approaches often cannot be further tuned or controlled by the human operator, limiting the applicability of the generated data to the target tasks. To mitigate these challenges, we introduce a novel modular and automated system named SIMFOUNDRY, enabling zero-shot real-to-sim scene construction from a single image or video. Our system supports automated object, scene, and task editing, enabling the generation of infinite variations of the real-world scene for training more generalizable policies. We leverage our system in both training (real-to-sim-to-real) and evaluation (real-to-sim) settings and show that our system can generate useful training data, as well as sim-ready scenes that produce correlative signals to real-robot evaluations of trained policies, resulting in more robust co-trained policies across a broad range of robot manipulation tasks, including multiple tasks that surpass the complexity of those shown by prior work and require multiple steps, articulated interaction, and bimanual coordination.**

*Equal contribution. †Equal advising.
[1]NVIDIA, [2]Georgia Institute of Technology, [3]Stanford University, [4]The University of Texas at Austin, [5]University of Toronto.

I. INTRODUCTION

Robotic foundation models [1], [2] trained on large-scale robot manipulation datasets have enabled robots to perform a wide range of manipulation tasks autonomously. However, sourcing high-quality robot manipulation data in large volumes is costly, and often involves robot teleoperation carried out by large teams of human operators over many months or years [1], [3]–[5]. Moreover, evaluating trained foundation models in a systematic and scientific manner on real-world manipulation problems of interest can require inordinate amounts of human time and effort, often requiring thousands of trials across different tasks to make rigorous comparisons [6].

In response to these critical bottlenecks, recent works have explored simulation as a potential viable alternative to training and evaluating robot manipulation models at scale. Automated data generation tools have demonstrated the possibility of synthesizing large volumes of diverse and high-quality robot manipulation demonstrations in simulation with minimal human effort [7]–[11], and have been used to train and improve real-world manipulation agents [12]–[15]. Furthermore, recent work [16], [17] has shown that model evaluations conducted in simulation can strongly correlate with real-world evaluation results, offering a time- and cost-efficient alternative to benchmarking policies requiring minimal human effort. However, constructing simulation environments manually can be challenging and tedious, especially when they must be carefully aligned with real-world scenes and tasks in terms of visuals, geometry, and dynamics.

To address the limitations of traditional simulation design, real-to-sim scene construction [18], [19] has emerged as a viable paradigm for generating synthetic scenes grounded in the real world. By leveraging 3D reconstruction and generative models to process real-world visual data, researchers can create environments with minimal manaul effort that are "sim-ready" – digitized scenes that support physically grounded robotic interaction. While this approach greatly reduces the overhead of environment authoring and enables both sim-to-real transfer [18]–[23] as well as more predictive real-world benchmarking [24]–[26], few works have showcased unified methods that both automate the scene reconstruction process and train compelling policies that transfer across domains. Recent works highlighting real-to-sim reconstruction capabilities [27]–[29] often show limited sim-to-real experiments, whereas works emphasizing sim-and-real benchmarking often focus on shorter-horizon, atomic type tasks and rely upon manually-tuned simulation scenes [22]–[24], [30], [31].

To this end, we introduce SIMFOUNDRY, a unified approach enabling fully automated zero-shot real-to-sim scene construction from a single image. SIMFOUNDRY supports automated object, scene, and task editing, enabling the generation of infinite variations of the real-world scene for generating large-scale, diverse synthetic data. SIMFOUNDRY environments can also be used to benchmark policies trained in the real world, and show salient correlation between our real-world and simulation evaluation results. Finally, policies can be trained with trajectory data from simulated SIMFOUNDRY environments and deployed successfully in their real-world counterparts. Unlike prior methods [18], [21]–[23], SIMFOUNDRY further enhances the generalization capabilities of these policies by training them on diverse data from generated environment variations, enabling deployment in scenes different from those in real-world video (*e.g.*, changing objects), and we showcase SIMFOUNDRY's real-to-sim and sim-to-real applicability over a diverse set of manipulation tasks whose complexity surpasses those proposed by previous efforts, including tasks requiring multiple steps, articulated interaction, and bimanual coordination.

Summary of Contributions:

- We introduce SIMFOUNDRY, a novel simulation environment generation system that is fully automated and modular and supports both rigid-body and articulated object generation. SIMFOUNDRY exhibits strong zero-shot 3D-reconstruction fidelity that can be further tuned quickly by a human operator, and supports automated object, scene, and task variations once the initial scene is reconstructed.
- We apply SIMFOUNDRY to a broad set of manipulation tasks and showcase how SIMFOUNDRY can produce environments that provide correlative signals for real-world policies evaluated in sim.
- We provide a set of benchmark tasks highlighting greater complexity compared to prior works, spanning multiple types of multi-step manipulation, including bimanual coordination and articulated object interaction, across multiple robot embodiments, and showcase how SIMFOUNDRY can generate diverse synthetic data samples for training robust sim policies that can transfer to the real-world.
- We provide both qualitative and quantitative analysis of SIMFOUNDRY and show how its 3D reconstruction fidelity is competitive with other state-of-the-art 3D reconstruction methods and can be quickly tuned to refine the scene output.

II. RELATED WORK

3D Asset Generation and Alignment. 3D asset reconstruction and generation have evolved from retrieval-based alignments to sophisticated generative and articulated modeling. Traditional retrieval methods [34]–[37] focus on aligning CAD models from databases to single-view images, a concept recently extended by zero-shot approaches like ACDC [20] and DiffCAD [38]. In parallel, generative frameworks have advanced static object creation. Recent high-fidelity object-centric reconstruction models [39]–[45] push the boundaries of asset quality. To achieve precise pose alignment in multi-object layouts, recent vision foundation models offer robust priors that enhance spatial realism, including depth cues [46]–[48], segmentation [49], [50], object pose and scale estimation [51]–[53]. Addressing the need for functional digital twins, research has significantly expanded from rigid into articulated objects. Building on earlier works like Shape2Motion [54] and RPM-Net [55], state-of-the-art methods [56]–[63] automate the generation of diverse, movable assets for unseen articulated objects. Our SIMFOUNDRY is inherently modular, leveraging a suite of primitive 3D asset generation and composition techniques for real2sim scene creation that ensures long-term adaptability. This architecture allows for the seamless integration of state-of-the-art tools as they emerge, while providing a mechanism to augment or refine outputs through user interventions.

Real-to-Sim for Simulation Environment Creation and Applications. The advent of high–quality 3D reconstruction methods and generative models for 3D synthesis has recently enabled several systems that seek to partially or completely automate simulation environment creation by capturing real–world scenes [18]–[20], [22]–[29], [31], [57], [64]–[68]. Some prior work [18], [20], [21], [32], [69], [70]

TABLE I: (System Comparison table): SIMFOUNDRY provides a unified and modular pipeline for real-to-sim scene generation that is more feature complete than other pre-existing works.

	ACDC [20]	RialTo [18]	DRAWER [32]	RoLA [21]	R2R2R [33]	SIMPLER [16]	Polaris [24]	RobotArena-∞ [26]	R2S-Soft [25]	Re^{3}Sim [22]	GSWorld [23]	SAGE [27]	MolmoSpaces [31]	GenieSim [30]	SIMFOUNDRY (Ours)
Sim-to-Real Training	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	✓	✓
Real-to-Sim Policy Eval	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓
Automatic Scene Construction	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓
Articulated Objects	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓
Multi-Embodiment	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓
Asset Generation	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓
Background Reconstruction	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓
Object Variations	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
Scene Variations	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
Task Variations	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓

constructs real–to–sim–to–real pipelines, digitizing physical scenes into simulation to train agents for real-world deployment. However, these systems are typically restricted to simple pick-and-place tasks; in contrast, SIMFOUNDRY supports a broader range of manipulation, including bimanual setups, interactions with articulated objects, and multi–step manipulation. Furthermore, while existing pipelines often focus on static digital twins—limiting environmental diversity and hindering generalization—SIMFOUNDRY generates multiple types of environment variations, including object, scene, and task variations. This increased diversity enables SIMFOUNDRY to facilitate more robust generalization across varied real-world conditions. Similar real–to–sim–to–real approaches have also been developed and deployed for robot navigation [71] and locomotion [72]. Some works circumvent the need for simulation [33] by using the reconstructed scenes purely for rendering, but this makes it difficult to apply to higher-precision tasks requiring accurate modeling of physics.

Another body of work focuses on using reconstructed simulation environments as a way to reliably evaluate manipulation policies [16], [17], [22]–[26], [30], such that the results strongly correlate with corresponding real-world evaluations. However, unlike SIMFOUNDRY, several such systems do not show that data from their simulation environments can be used to train real–world agents [16], [17], [24]–[26], and to our knowledge, no systems support tasks involving general articulated objects. SIMFOUNDRY belongs to a select group of real–to–sim systems [22], [23], [30] that demonstrate both successful sim-to-real agent transfer and strong correlation between simulated and physical policy evaluations. However, SIMFOUNDRY goes beyond these systems, in its ability to handle more diverse task characteristics (including bimanual, articulation, and multi–step manipulation) and its support for developing multiple types of variations of a digitized real–world scene to scale up the diversity of reconstructed environments (see Table I for a summary).

Imitation Learning from Human Demonstrations and Synthetic Data Generation. Robot teleoperation [73], [74] is a common approach for collect demonstrations to train robots to perform manipulation tasks autonomously – here, a human uses a teleoperation device (such as a smartphone or a VR controller) to guide a robot through different tasks, and the resultant robot sensor streams and actions are logged to a dataset. Robot manipulation policies are often trained on such datasets with Behavioral Cloning (BC) [75]–[79]. In recent years, this approach has been scaled up to collect months of data using large teams of human operators [3]–[5], [80], and has proved to be very effective for robot manipulation [1], [5], [81], [82]. However, data collection is a bottleneck, since it is time–consuming and expensive. A recent line of work leverages synthetic data generation (SDG) in simulation [7]–[10], [83] as a compelling alternative to address the need for large-scale datasets. Recent evidence has shown that these synthetic datasets can supplement or even replace real-world datasets to reduce the burden of real-world data collection [2], [12]–[15], [30], [84]. We use such tools to highlight an important application of our system – real–to–sim–to–real policy learning. Here, we reconstruct a simulation environment (along with controlled variations) from a real-world environment, generate synthetic data in simulation, and train manipulation agents that transfer to the real-world, all with minimal human effort.

III. PRELIMINARIES

Overview. In this work, we seek to apply SIMFOUNDRY to first reconstruct real world scenes $\mathcal{S}_{real}$ in simulation $\mathcal{S}_{sim}$ by converting an input image $\mathbf{I}_s$ into a set of object meshes $\mathcal{M}_i$, scales $\mathbf{s}_i$, and poses $\mathbf{p}_i$, where $i \in \{1, ..., N\}$, leveraging multiple foundation models V_* to achieve this. We then apply SIMFOUNDRY to downstream robotics applications, including real-to-sim evaluation and real-to-sim-to-real training of robot policies. We broadly define a policy π_θ mapping observations at the current timestep o_t to action chunk over time horizon H a_t^{t+H}, $\pi_\theta : \mathcal{O} \rightarrow \mathcal{A}$, and implemented as a neural network parameterized by θ. In practice, $\mathcal{O}$ may contain multiple modalities of observations,

including language conditioning o_{lang}, images o_{img}, and proprioception $o_{proprio}$.

Real and Simulation Policy Correlation. We measure real and sim policy correlation using the Pearson Correlation Coefficient (r) and Mean Maximum Rank Violation (MMRV), both of which have been proposed by prior works [16], [24]. Ideal correlation has $r \rightarrow 1$, which measures linear correlation between real and simulation task results, and MMRV $\rightarrow 0$, which measures the average worst rank-violation of policies as evaluated in simulation versus their actual ranks as evaluated in the real world. When measuring task success, we measure both end-to-end task success (a discrete 0 or 1) as well as normalized task reward (continuous value between 0 and 1) as quantitative metrics.

Sim-to-Real Training. We define sim-to-real transfer as deploying a policy trained exclusively in simulation zero-shot in the real world. Successful transfer is indicated by non-zero real-world success and minimal degradation from simulation performance. Task success is measured by normalized reward in $[0, 1]$, where 1 denotes full task completion.

Data Augmentation. We define *digital twins* as being strict replicas of the geometry and object layouts of a real-world scene. In contrast, *digital cousins* [20] are virtual scenes that maintain the semantic and geometric affordances of a real-world scene without explicitly modeling it, and serve as a form of object instance randomization. MimicGen [9] is a recent method proposed for quickly generating large amounts of synthetic data by splicing together various subtask trajectories sampled from a set of source demonstrations, and utilizes rejection sampling to only preserve the successful demonstrations. Co-training [12] is a method proposing augmenting training data to include both sim and real data, in order to induce more robustness in both domains. For real-to-sim evaluation, this ideally improves correlation, while for sim-to-real training, this tends to improve domain transfer robustness.

IV. SimFoundry: A modular, automated real-to-sim generation pipeline

SimFoundry is a modular, end-to-end system for generating diverse fully interactive simulated scenes. Our pipeline is composed of a sequence of three steps: (1) an **Extraction** process that infers per-object relevant information from a raw input video or image, (2) a **Generation** process that creates annotated sim-ready object meshes aligned to the original scene, and (3) a **Variation** process that augments the initial simulated scene and enables diverse scene configurations. Below, we briefly describe each component of our system. For additional details, including specific off-the-shelf foundation models (denoted as V_*) supported by our pipeline, please see Appendix F.

Extraction. We first extract raw vision modalities necessary for generating and aligning individual objects to the original scene. We assume the input is either a raw video of the scene, a single RGB image, or a paired RGB stereo image. We first convert any of these inputs into a single representative RGB frame $\mathbf{I}_s$ for the scene and extract its corresponding synthetic depth map $\mathbf{D}_s$ using off-the-shelf depth estimation models $V_{im2depth}$. Then, using the camera intrinsics $\mathbf{K}$, a point cloud $\mathbf{P}_s$ of the scene is generated. Next, we query V_{seg}^{image} [41] to detect and extract a ground plane that will be used to align the scene with the simulator world frame. Then, we iteratively query a scene understanding vision-language-model (VLM) V_{scene} to decompose the scene by detecting foreground object categories $\{o_1, o_2, \ldots, o_n\}$ and adding point prompts to each detected object. For each iteration $i \in \{1, 2, \ldots, n\}$, V_{seg}^{image} outputs the detected object category's corresponding segmentation mask m_i, which is used to extract the rgb p_i^{rgb} and depth p_i^{depth} pixels for that object. The segmentation mask is passed to an image inpainting model V_{image} to remove the object from the scene image. The depth pixels for that object are set to zero, $\bar{D}_i = D_{i-1} \odot (1 - m_i)$, where D_{i-1} is the depth map before o_i is removed, and then the missing values filled in by a depth inpainting model $V_{inpaint}^{depth}$, $D_i = V_{inpaint}^{depth}(\bar{D}_i)$. This cycle is repeated until V_{scene} no longer detects any foreground objects.

Generation. Given the per-object rgb pixels p_i^{rgb}, we first pad these pixels to a default size, then use V_{image} to upsample this padded image. We pass the upsampled image to a 2D-to-3D mesh model V_{mesh} to generate the object's visual mesh $\mathcal{M}_i$. We then align this mesh to the original scene by first coarsely estimating its $\mathbf{s}_i$ and pose $\mathbf{p}_i$ using coherent point drift [85] applied to the point cloud representation of the mesh, and then further refine the pose $\mathbf{p}_i$ using $\mathbf{I}_s$, $\mathbf{D}_s$, m_i and $\mathcal{M}_i$ with FoundationPose [51]. Objects that are detected by V_{scene} as being articulated (such as cabinets, drawers) are passed through a separate module which builds upon prior methods [63], [86] to generate joint parameters in a fully automated way: first querying a VLM $V_{articulation}$ to detect parts of the object that are articulated such as doors and knobs, then segmenting $\mathcal{M}_i$ using a mesh segmentation model V_{seg}^{mesh}. The joint parameters linking these segmented parts are generated via $V_{articulation}$ (further details in Appendix G). For each generated object, we then generate collision representations of the object using CoACD [87] and physics annotations (mass, friction) by querying V_{scene}. Once all objects are generated, aligned, and annotated, we compose and spawn the final scene in PyBullet [88] and allow any overlapping objects to slowly depenetrate, guaranteeing a physically stable scene. Our final composed scene can then be exported to a purpose-built robotics simulator such as IsaacSim and immediately used out-of-the-box.

Variation. Once the initial generated scene is constructed, we provide 3 types of variations that can be applied in an automated way to diversify the initial outputs, described briefly below.

Object variations: Given a single input object, we employ a three-stage pipeline to generate multiple plausible variations of the same object, conditioned on its affordance. First, given the original object image, we prompt V_{image} to decompose the object into parts based on action af-

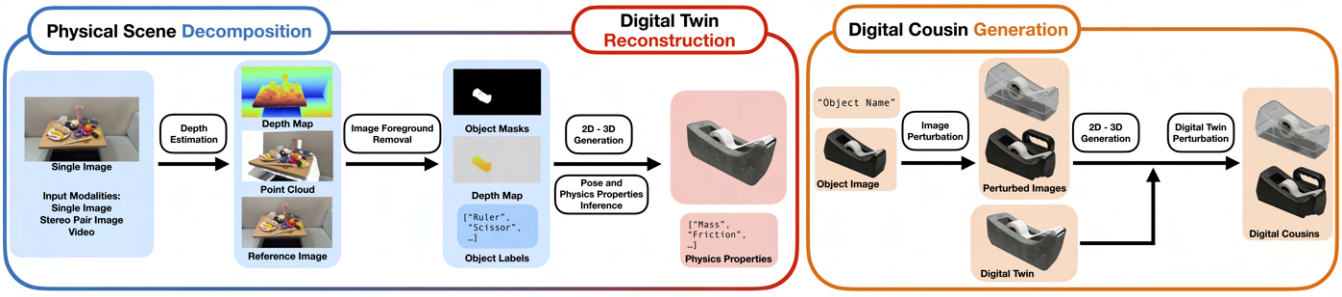


Fig. 2: **Method Overview.** SIMFOUNDRY first extracts per-object relevant information (segmentation masks, depth, etc.), generates 3D visual meshes via 2D-to-3D generation models, and compiles the final output scene by annotating relevant physical parameters and sanity checking the overall scene configuration in a physics simulator. Finally, additional scene variations can be produced by modifying the scene geometry in 2D space, which is then propagated to novel 3D generated shapes that populate the new scenes.

fordance. This is motivated by prior work on affordance-based decomposition [89], [90] and morphological analysis [91], which demonstrate that objects can be decomposed into functional components and systematically varied along independent dimensions. For each decomposed part, we then prompt V_{image} to propose reasonable variations along multiple axes as a line of text prompt, including the part's geometry (e.g., aspect ratios, curvature profiles), topology (e.g., handle configurations, structural connectivity), and visuals (e.g., surface texture, material properties). These axes allow us to vary the object while maintaining the object category's recognizability and functionality. Finally, we feed each proposed variation text prompt and the target object image into V_{image} to generate a modified version of the object image, which is then processed by V_{mesh} to generate a new 3D object mesh.

Scene variations: Starting from the canonical spatial arrangement of objects in the generated scene, we apply randomization to vary their relative placements. For instance, if the reconstructed scene has a spoon placed to the right of a plate, we vary the spoon's placement to have to be placed on top of, or to the left of the plate. These variations are specified using spatial predicates such as **OnTop** or **RightOf** . Additionally, we select N distractor objects from a dataset of sim-ready assets to add into our scenes. The types and physical properties of the distractor objects can be controlled to ensure they conform to the scene. These scene-level variations help to train more robust policies by injecting meaningful and controlled geometric domain randomization.

Task variations: Given the sim-ready reconstructed scene, we query V_{scene} to propose feasible manipulation tasks grounded in the environment. The VLM acts as an automated task generator by outputting configuration files that encapsulate precise goal conditions and success criteria using predicate language and can be directly deployed in simulation. With this pipeline, we can then procedurally collect expert demonstrations for the proposed tasks to train a multi-task policy. This coupling of automated task generation and simulation eliminates the need for manual scene configuration, offering a method to scale up multi-task manipulation data.

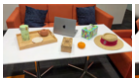
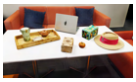
N Objects	Wall Clock Time (s)	Time per Object (s)	Original Image	Reconstructed Twin	Cousins Image
10	2565.95	256.595			
10	3083.37	308.337			
10	3825.44	382.544			
9	2615.11	290.570			
9	2481.70	275.740			
10	2941.21	294.121			

TABLE II: **Real-to-sim reconstruction results.** We show real-world scene input images, alongside SIMFOUNDRY's generated output and a sampled object variation instance as well as corresponding wall-clock time measurements for running our pipeline.

V. EXPERIMENTS

Sec. V-A describes our experiment setup. We highlight two key applications of SIMFOUNDRY – training robot manipulation agents from generated SIMFOUNDRY environments that transfer zero-shot to the real-world (Sec. V-B), and using SIMFOUNDRY environments as a way to benchmark real-world manipulation policies (Sec. V-C). Sec. V-D contains additional experiments that analyze SIMFOUNDRY performance.

A. Experimental Setup

Robot Embodiments. We focus on two robot embodiments – the DROID [5] platform, and a YAM workcell [92]. The DROID platform consists of a single Franka Panda robot arm, left and right external ZED-2 cameras, and a wrist-mounted ZED-Mini camera, with the robot and external

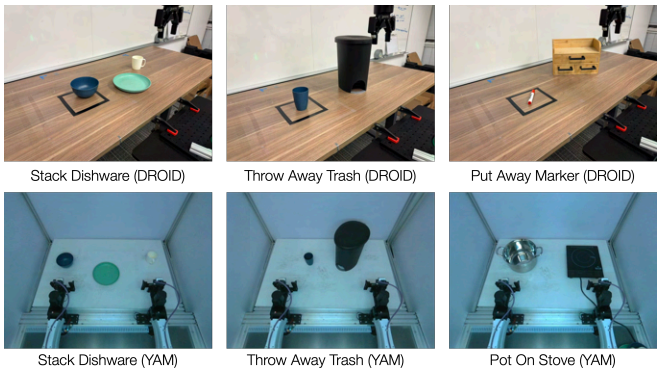


Fig. 3: **Tasks.** Our in-the-wild tasks in real deployed on two embodiments, a DROID setup using a single Franka arm (top), and a bimanual setup with two YAM arms (bottom). Our tasks span multiple types of manipulation, including multi-step, articulated object interaction, and bimanual coordination. For additional details, please see Appendix D.

cameras mounted to a portable standing desk. The YAM workcell consists of a bimanual manipulator, a cage, a wrist-mounted RGB camera per arm, and a top-down view camera.

Tasks. Our tasks span pick-and-place and articulated object interaction, and includes both single and multi-step manipulation (Fig 3). Our task suite (and corresponding success) on DROID is:

- **Cup in Bowl** : A paper cup and a large bowl are placed on the table. The cup must be picked up and placed in the bowl. This is our easiest task and serves to validate that easily solved tasks are similarly easy in our sim.
- **Marker in Cup** : A marker and a cup are placed on the table. The marker must be picked up and placed in the cup. This task tests action precision and the fidelity of object sizes and shapes.
- **Serve Fruits** : Two plates, one orange and one green are placed on the table, along with a variety of fruits. The banana and the apple must be placed on the green plate. This task tests language following and manipulation in clutter.
- **Store Marker** : A small cabinet and marker are placed on the table. After opening the drawer, the marker must be placed into the drawer, after which the drawer must finally be closed again. This task showcases our ability to recreate scenes with articulated objects, and evaluate policies on them.

The following tasks are tested on both YAM and DROID:

- **Stack Dishware** : A bowl, plate, and mug are placed on the table. The bowl must be placed on top of the plate and the mug placed on top of the bowl without tipping. This task combines long-horizon behavior with precision and language following.
- **Put Away Trash** : A trashcan and cup are placed on the table. After opening the lid, the cup must be placed into the bin, after which the lid must be closed again.

Finally, on YAM, we evaluate another task involving bimanual coordination:

- **Pot on Stove** : A two-handled pot and stove are placed on the table. The pot must be lifted by both arms and placed stably on the stovetop. This challenging task requires both bimanual coordination and precise grasping.

Real-to-Sim-to-Real Policy Training. We aim to show that manipulation policies trained with data from SIMFOUNDRY environments can transfer to the real-world, with high-performance, and further, that the agents can generalize to unseen real-world conditions. To this end, we first reconstruct real-world scenes with SIMFOUNDRY, and then apply our different types of variations to further augment the set of generated scene instances.

We then collect 4 to 15 human demonstrations for each reconstructed scene instance, and then generate 100 to 1000 synthetic demonstrations with MimicGen [8] and DexMimicGen [9], with additional visual domain randomization applied. Using this data, we then train multiple types of RGB-based policies, including flow-matching policies [93] trained from scratch and VLA models [1], [2] finetuned from pretrained checkpoints.

Simulation-based Policy Evaluation Setup. We aim to show that real-world policy evaluation rankings in SIMFOUNDRY can correlate well with real-world evaluations. To this end, we first reconstruct real-world scenes with SIMFOUNDRY and then evaluate policy models in both the original real-world scene and the reconstructed scenes. Concretely, we test the following pretrained VLA models:

- $\pi_{0.5}$: the publicly released version of $\pi_{0.5}$ [94] fine-tuned on the DROID dataset.
- π_0: the π_0 model fine-tuned on the DROID dataset.
- N1.6: the publicly released DROID-specific version of the Gr00t series of models [2].

We evaluate all models on the DROID setup across the DROID-compatible tasks mentioned in Fig 3. The first three tasks [**Cup in Bowl** , **Marker in Cup** , **Serve Fruits**] are evaluated zero-shot whereas for the rest [**Stack Dishware** , **Store Marker** , **Put Away Trash**], we collect 50 finetuning demos per task in the real world, since these cannot be solved easily by the base VLAs.

Evaluation Process. On both YAM and DROID, each task is divided into subtasks and evaluated for 25 rollouts per policy both in sim and the real world, with aligned scene initializations. The object poses for each of the 25 rollouts are kept consistent across policies. Each rollout is scored from 0 to 1.0 based on the proportion of subtasks completed, with each subtask weighted equally. The subtasks are scored independently and do not necessarily have to be completed in sequence. For example, for the **Stack Dishware** task, the policy can get the reward for placing the cup on the bowl, even if the bowl is not placed on the plate. Equivalent scoring is applied and automated in simulation. Further details, including the scoring criteria for each task, are listed in Appendix A. For the real-to-sim evaluation of the fine-tuned tasks, we also conduct a subtask-based evaluation in simulation. We do so by initializing each episode from an

TABLE III: **Real-to-sim evaluation results – full success rates on DROID.** Success rates are reported as percentages for real-world evaluations (R) and reconstructed-scene simulations (S) across six tasks. For each task, Pearson correlation r (↑) and MMRV (↓) are computed across models.

Task	π_0		$\pi_{0.5}$		N1.6		Metrics	
	R	S	R	S	R	S	r ↑	MMRV↓
Cup in Bowl	88	56	100	92	68	40	0.94	0.00
Marker in Cup	40	40	92	88	28	28	1.00	0.00
Serve Fruits	0	4	72	80	4	20	0.99	0.00
Stack Dishware	100	34	100	64	40	0	0.88	0.00
Store Marker	48	4	60	20	32	0	0.91	0.00
Put Away Trash	20	0	48	4	0	0	0.91	0.07

TABLE IV: **Real-to-sim evaluation results - subtask success rates (DROID).** Evaluations on subtasks in sim can improve correlation for long-horizon tasks finetuned on a single scene.

Task	π_0		$\pi_{0.5}$		N1.6		Metrics	
	R	S	R	S	R	S	r ↑	MMRV↓
Stack Dishware	100	64	100	80	40	24	0.96	0.00
Store Marker	48	52	60	76	32	36	0.98	0.00
Put Away Trash	20	0	48	8	0	0	0.91	0.06

intermediate state in which the initial portion of the task has already been completed. Concretely, for **Stack Dishware** we start from states where subtasks 1 and 2 are complete, while for **Store Marker** and **Put Away Trash** we start from states where subtask 1 is complete.

TABLE V: **Sim-to-real policy training results (YAM)**

		Twin	+ 9 Cousins
Stack Dishware	Sim Twin	83	92
	Sim Cousins	43	66
	Real Twin	39	43
	Real Cousins	21	42
Pot On Stove	Sim Twin	85	100
	Sim Cousins	17	93
	Real Twin	71	72
	Real Cousins	-	-
Throw Away Trash	Sim Twin	97	97
	Sim Cousins	97	94
	Real Twin	0	28
	Real Cousins	2	8

B. Real-to-Sim-to-Real Policy Training

Policies trained with synthetic data from reconstructed SIMFOUNDRY environments can transfer zero-shot to their real-world counterparts. All the policies evaled on the YAM setup are able to perform the tasks on the real world objects we reconstruct in sim, with success rates reaching as high as 100% for the Pot On Stove task. Additionally, we see increasing performance, especially on unseen objects, when policies are trained with simulated cousins. Thus, SIMFOUNDRY supports a real-to-sim-to-real policy learning workflow, where a real world scene can be reconstructed, policies can be trained in simulation, and then deployed zero-shot in the original real world scene.

Policies trained with synthetic data from diverse SIMFOUNDRY environments generalize to novel real-world conditions, including previously unseen rigid and articulated objects. As shown in Table V, policies trained with digital cousins tend to be much more robust when deployed in the real world, both on the original target object(s) as well as held-out, unseen real-world objects. Notably, these objects were never explicitly modeled or reconstructed as simulation assets. Instead, SIMFOUNDRY generates a suite of digital cousins via natural language, providing the environmental diversity required for real-world generalization. By training on this diverse distribution of synthesized assets, agents develop the robustness necessary to handle novel objects.

C. Simulation-based Policy Evaluation

Policy rankings correlate well between SIMFOUNDRY environments and the real-world across tasks with different characteristics, including articulated object manipulation. As shown in Table III, we find that task success rates show close correlation between sim and real, especially in tasks without finetuning. This highlights SIMFOUNDRY's potential to generate salient virtual environments that can be immediately used to evaluate real-world policies in a zero-shot manner. We find that our findings hold even for difficult tasks with articulated objects (Table IV), which have not been covered by prior work. Relative rankings between policies are maintained between sim and real, which is demonstrated by our low MMRV scores.

Policy evaluation correlations degrade on longer-horizon tasks finetuned in a single scene, but subtask-based evaluation can still provide actionable insights. For longer-horizon finetuned tasks, absolute success rates may correlate poorly with real-world performance due to compounding distribution shift. To our knowledge, this is the first work to characterize the degradation of standard correlation metrics on longer-horizon tasks. We attribute this to small scene-alignment errors, which are especially consequential for finetuned models; even lighting or camera-pose changes in the same real-world scene can affect policy performance [95]. Subtask-based simulation evaluation mitigates this issue by decomposing harder tasks into shorter segments, yielding non-zero success rates and more informative diagnostics of policy quality and failure modes.

D. System Analysis

In this section, we analyze several characteristics of SIMFOUNDRY in greater detail, including the fidelity of automatically reconstructed environments, the amount of human effort needed to improve their fidelity further, and the scalability of the environment generation pipeline. We first measure the efficiency and qualitative outputs of our system by recording multiple in-the-wild scenes and reconstructing

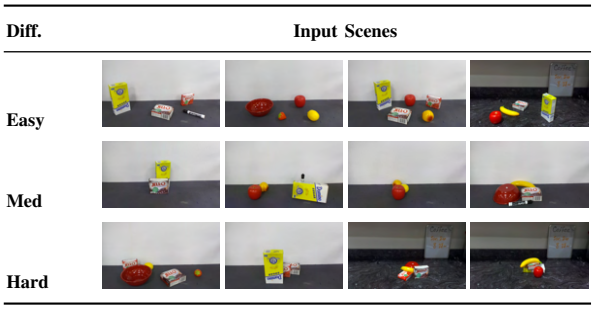


Fig. 4: **Reconstructed Scenes.** Reconstructed scenes categorized by Easy (No Occlusion), Mid (Slight Occlusion), and Hard (Strong Occlusion).

them using our automated pipeline. The resulting wall-clock time and corresponding outputs can be seen in Table VII.

We then assess the quantitative reconstruction capabilities of our system by reconstructing 12 physically staged real-world environments with varying degrees of clutter and occlusion, as shown in 4, and evaluate our method against SAM3D [96], a state-of-the-art method for end-to-end 3D scene reconstruction from a single image. Concretely, we stage scenes of increasing difficulty (clutter / occlusion) using objects from the YCB dataset [97], and measure reconstruction fidelity via 3D geometric metrics: Chamfer Distance, F1-Score and Object Bounding Box Position Error. We additionally measure the reconstruction fidelity by allowing an operator to briefly tune each outputted object's scale and pose using one of our interactive scripts (see Appendix E). Additional details can be found in Appendix C. Our results are shown in Table VI.

SIMFOUNDRY environment reconstruction fidelity outperforms other state-of-the-art methods. As shown in Table VI, we find that our system under zero-shot (fully automated output) conditions generally outperforms SAM3D across all metrics over all scenes, showcasing SIMFOUNDRY's ability to reconstruct scenes with high geometric fidelity in a fully automated manner.

TABLE VI: Quantitative Reconstruction Results (Average $\pm$ Standard Deviation)

Difficulty	Metric	SAM3D Zero Shot	CDC Zero Shot	CDC Tuned (3min/Obj)
Easy	Chamfer Dist (m) ↓	0.0081 ± 0.0024	0.0042 ± 0.0013	$\mathbf{0.0026 \pm 0.00026}$
	F1 Score ↑	0.71 ± 0.15	0.92 ± 0.071	$\mathbf{0.99 \pm 0.0069}$
	Pos Error (m) ↓	0.016 ± 0.0058	0.0060 ± 0.0019	$\mathbf{0.0041 \pm 0.00037}$
Mid	Chamfer Dist (m) ↓	0.0087 ± 0.0028	0.0047 ± 0.0012	$\mathbf{0.0033 \pm 0.00068}$
	F1 Score ↑	0.66 ± 0.18	0.87 ± 0.089	$\mathbf{0.97 \pm 0.026}$
	Pos Error (m) ↓	0.018 ± 0.0067	0.0076 ± 0.0038	$\mathbf{0.0057 \pm 0.0030}$
Hard	Chamfer Dist (m) ↓	0.0088 ± 0.0022	0.0091 ± 0.0076	$\mathbf{0.0039 \pm 0.0013}$
	F1 Score ↑	0.68 ± 0.14	0.81 ± 0.071	$\mathbf{0.93 \pm 0.049}$
	Pos Error (m) ↓	0.022 ± 0.010	0.018 ± 0.018	$\mathbf{0.0073 \pm 0.0022}$

SIMFOUNDRY environment generation scales well with compute and human time. As shown in Table VII, we find that our system can reconstruct a broad set of objects and scenes, and can run at an average rate of 5 minutes per object, highlighting SIMFOUNDRY's ability to generalize to many real-world scenes with tractible compute time. Moreover, as seen in Table VI, we find that with only 3 additional minutes of tuning per object, the 3D reconstruction metrics reliably improve even further over SIMFOUNDRY's already performant output, highlighting our system's modular ability to quickly and iteratively tune the reconstructed scene based on a user's fidelity requirements.

VI. LIMITATIONS

Our system relies heavily upon off-the-shelf foundation models. While this enables broad modularity and the ability to swap models on the fly, it does naturally incur the limitations of the utilized foundation models, including all of its quirks and failure modes. All of the VLMs used in this paper are queried remotely via a 3rd party provider, which can cause indeterminism across identical runs. For example, we find that the inpainting of Gemini image models to be occasionally inconsistent, leading to degenerate extracted objects. Moreover, 3D reconstruction fidelity is also heavily reliant upon the underlying inferred point cloud fidelity. For monocular image inputs, this can reduce the accuracy of the outputted scene, where the scale and shape of the synthetic point cloud may not completely match the corresponding real-world scene. Our articulation results depend on accurate 3D segmentation of the object mesh, which can be difficult for meshes generated by image-to-mesh models or those with internal structures. Physics stability also implicitly assumes that objects rest upon a flat surface, which generally restricts our pipeline to table-top setups.

When reconstructing scenes for policy evaluation and training in simulation, we also need to manually scan the background and align the resulting background mesh to the foreground scene and robot, as well as manually tune each reconstructed object's scale further so that they match their real-world corresponding objects' dimensions. This stems, in part, from the underutilization of information from the input video scene, and future iterations of our pipeline would ideally infer both the background and the metric object scale more precisely by leveraging this information.

VII. CONCLUSION

SIMFOUNDRY is a fully automated pipeline that natively reconstructs interactive sim-ready scenes from a single image input, handles articulated object generation and scenes with clutter and occlusion, and generates object, scene, and task variations. We find that SIMFOUNDRY can measure task success in simulation that correlates to real-world policy performance, and can also be used to generate large amounts of synthetic data that can be used to train robot manipulation policies that transfer zero-shot to the real world.

ACKNOWLEDGMENTS

The authors would like to thank Omkaar Buddhikot, Amitoj Sandhu, Nadia Laswi, Ramanpreet Singh, Mona Abbas, and Osiriz Durana for their help with data collection and model evaluation. We also thank Jeremy Chimienti, Danyi Chen, and Lion Park for their help with hardware support, and Scott Reed, You Liang Tan, and Fengyuan Hu for feedback and valuable discussions. Nadun Ranawaka is partially supported by the Agricultural Technology Research Program at the Georgia Institute of Technology.

REFERENCES

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[2] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[3] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets," in *Robotics: Science and Systems*, 2022.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[5] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[6] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina *et al.*, "A careful examination of large behavior models for multitask dexterous manipulation," *arXiv preprint arXiv:2507.05331*, 2025.

[7] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," in *Conf on Robot Learning*, 2023.

[8] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[9] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," *arXiv preprint arXiv:2410.24185*, 2024.

[10] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," *arXiv preprint arXiv:2410.18907*, 2024.

[11] C. Li, M. Xu, A. Bahety, H. Yin, Y. Jiang, H. Huang, J. Wong, S. Garlanka, C. Gokmen, R. Zhang, W. Liu, J. Wu, R. Martín-Martín, and L. Fei-Fei, "Momagen: Generating demonstrations under soft and hard constraints for multi-step bimanual mobile manipulation," in *RSS 2025 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2025. [Online]. Available: https://openreview.net/forum?id=4ATOUj1k9n

[12] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev *et al.*, "Sim-and-real co-training: A simple recipe for vision-based robotic manipulation," *arXiv preprint arXiv:2503.24361*, 2025.

[13] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake, "Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels," *arXiv preprint arXiv:2503.22634*, 2025.

[14] S. Cheng, L. Ma, Z. Chen, A. Mandlekar, C. Garrett, and D. Xu, "Generalizable domain adaptation for sim-and-real policy co-training," *arXiv preprint arXiv:2509.18631*, 2025.

[15] S. Haldar, L. Johannsmeier, L. Pinto, A. Gupta, D. Fox, Y. Narang, and A. Mandlekar, "Point bridge: 3d representations for cross domain policy learning," *arXiv preprint arXiv:2601.16212*, 2026.

[16] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[17] A. Badithela, D. Snyder, L. Zha, J. Mikhail, M. O'Kelly, A. Dixit, and A. Majumdar, "Reliable and scalable robot policy evaluation with imperfect simulators," *arXiv preprint arXiv:2510.04354*, 2025.

[18] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal, "Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation," *arXiv preprint arXiv:2403.03949*, 2024.

[19] P. Dan, K. Kedia, A. Chao, E. W. Duan, M. A. Pace, W.-C. Ma, and S. Choudhury, "X-sim: Cross-embodiment learning via real-to-sim-to-real," *arXiv preprint arXiv:2505.07096*, 2025.

[20] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei, "Automated creation of digital cousins for robust policy learning," *arXiv preprint arXiv:2410.07408*, 2024.

[21] S. Zhao, J. Mao, W. Chow, Z. Shangguan, T. Shi, R. Xue, Y. Zheng, Y. Weng, Y. You, D. Seita *et al.*, "Robot learning from any images," in *Conference on Robot Learning*. PMLR, 2025, pp. 4226–4245.

[22] X. Han, M. Liu, Y. Chen, J. Yu, X. Lyu, Y. Tian, B. Wang, W. Zhang, and J. Pang, "Re3sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation," *arXiv preprint arXiv:2502.08645*, 2025.

[23] G. Jiang, H. Chang, R.-Z. Qiu, Y. Liang, M. Ji, J. Zhu, Z. Dong, X. Zou, and X. Wang, "Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation," *arXiv preprint arXiv:2510.20813*, 2025.

[24] A. Jain, M. Zhang, K. Arora, W. Chen, M. Torne, M. Z. Irshad, S. Zakharov, Y. Wang, S. Levine, C. Finn *et al.*, "Polaris: Scalable real-to-sim evaluations for generalist robot policies," *arXiv preprint arXiv:2512.16881*, 2025.

[25] K. Zhang, S. Sha, H. Jiang, M. Loper, H. Song, G. Cai, Z. Xu, X. Hu, C. Zheng, and Y. Li, "Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions," *arXiv preprint arXiv:2511.04665*, 2025.

[26] Y. Jangir, Y. Zhang, K. Yamazaki, C. Zhang, K.-H. Tu, T.-W. Ke, L. Ke, Y. Bisk, and K. Fragkiadaki, "RobotArena ∞: Scalable robot benchmarking via real-to-sim translation," *arXiv preprint arXiv:2510.23571*, 2025.

[27] H. Xia, X. Li, Z. Li, Q. Ma, J. Xu, M.-Y. Liu, Y. Cui, T.-Y. Lin, W.-C. Ma, S. Wang, S. Song, and F. Wei, "Sage: Scalable agentic 3d scene generation for embodied ai," 2026.

[28] N. Pfaff, T. Cohn, S. Zakharov, R. Cory, and R. Tedrake, "Scenesmith: Agentic generation of simulation-ready indoor scenes," 2026.

[29] Z. Wang, Y. He, L. Yang, W. Zou, H. Ma, L. Liu, W. Sui, Y. Guo, and H. Su, "Tabletopgen: Instance-level interactive 3d tabletop scene generation from text or single image," 2025.

[30] C. Yin, D. Huang, D. Yang, J. Wang, N. Zhao, C. Xu, W. Sun, L. Hou, Z. Li, J. Wu *et al.*, "Genie sim 3.0: A high-fidelity comprehensive simulation platform for humanoid robot," *arXiv preprint arXiv:2601.02078*, 2026.

[31] Y. Kim, W. Pumacay, O. Rayyan, M. Argus, W. Han, E. VanderBilt, J. Salvador, A. Deshpande, R. Hendrix, S. Jauhri, S. Liu, N. M. M. Shafiullah, M. Guru, A. Eftekhar, K. Farley, D. Clay, J. Duan, A. Guru, P. Wolters, A. Herrasti, Y.-C. Lee, G. Chalvatzaki, Y. Cui, A. Farhadi, D. Fox, and R. Krishna, "Molmospaces: A large-scale open ecosystem for robot navigation and manipulation," 2026.

[32] H. Xia, E. Su, M. Memmel, A. Jain, R. Yu, N. Mbiziwo-Tiapo, A. Farhadi, A. Gupta, S. Wang, and W.-C. Ma, "Drawer: Digital reconstruction and articulation with environment realism," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 771–21 782.

[33] J. Yu, L. Fu, H. Huang, K. El-Refai, R. A. Ambrus, R. Cheng, M. Z. Irshad, and K. Goldberg, "Real2render2real: Scaling robot data without dynamics simulation or robot hardware," *arXiv preprint arXiv:2505.09601*, 2025.

[34] W. Kuo, A. Angelova, T.-Y. Lin, and A. Dai, "Mask2cad: 3d shape prediction by learning to segment and retrieve," in *European Conference on Computer Vision*. Springer, 2020, pp. 260–277.

[35] ——, "Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 589–12 599.

[36] C. Gümeli, A. Dai, and M. Nießner, "Roca: Robust cad model retrieval and alignment from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4022–4031.

[37] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2cad: Learning cad model alignment in rgb-d scans," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 2614–2623.

[38] D. Gao, D. Rozenberszki, S. Leutenegger, and A. Dai, "Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–15, 2024.

[39] T. H. Team, "Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details," 2025. [Online]. Available: https://arxiv.org/abs/2506.16504

[40] J. Xiang, X. Chen, S. Xu, R. Wang, Z. Lv, Y. Deng, H. Zhu, Y. Dong, H. Zhao, N. J. Yuan, and J. Yang, "Native and compact structured latents for 3d generation," *Tech report*, 2025.

[41] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer, "Sam 3: Segment anything with concepts," 2025.

[42] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," *arXiv preprint arXiv:2404.07191*, 2024.

[43] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, "Lrm: Large reconstruction model for single image to 3d," *arXiv preprint arXiv:2311.04400*, 2023.

[44] K. Wu, F. Liu, Z. Cai, R. Yan, H. Wang, Y. Hu, Y. Duan, and K. Ma, "Unique3d: High-quality and efficient 3d mesh generation from a single image," *Advances in Neural Information Processing Systems*, vol. 37, pp. 125 116–125 141, 2024.

[45] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, "Triposr: Fast 3d object reconstruction from a single image," *arXiv preprint arXiv:2403.02151*, 2024.

[46] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5249–5260.

[47] B. Wen, S. Dewan, and S. Birchfield, "Fast-FoundationStereo: Real-time zero-shot stereo matching," *arXiv preprint arXiv:2512.11130*, 2025.

[48] H. Lin, S. Chen, J. Liew, D. Y. Chen, Z. Li, G. Shi, J. Feng, and B. Kang, "Depth anything 3: Recovering the visual space from any views," *arXiv preprint arXiv:2511.10647*, 2025.

[49] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[50] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*. Springer, 2024, pp. 38–55.

[51] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[52] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon, "Any6d: Model-free 6d pose estimation of novel objects," *CVPR*, 2025.

[53] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.

[54] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2motion: Joint analysis of motion parts and attributes from 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8876–8884.

[55] Z. Yan, R. Hu, X. Yan, L. Chen, O. Van Kaick, H. Zhang, and H. Huang, "Rpm-net: recurrent prediction of motion and parts from point cloud," *arXiv preprint arXiv:2006.14865*, 2020.

[56] Y. Weng, B. Wen, J. Tremblay, V. Blukis, D. Fox, L. Guibas, and S. Birchfield, "Neural implicit representation for building digital twins of unknown articulated objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3141–3150.

[57] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5616–5626.

[58] J. Liu, D. Iliash, A. X. Chang, M. Savva, and A. Mahdavi-Amiri, "Singapo: Single image controlled generation of articulated parts in objects," *arXiv preprint arXiv:2410.16499*, 2024.

[59] C. Chen, I. Liu, X. Wei, H. Su, and M. Liu, "Freeart3d: Training-free articulated object generation using 3d diffusion," in *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025, pp. 1–13.

[60] Z. Chen, A. Walsman, M. Memmel, K. Mo, A. Fang, K. Vemuri, A. Wu, D. Fox, and A. Gupta, "Urdformer: A pipeline for constructing articulated simulation environments from real-world images," *arXiv preprint arXiv:2405.11656*, 2024.

[61] S. Yuan, R. Shi, X. Wei, X. Zhang, H. Su, and M. Liu, "Larm: A large articulated object reconstruction model," in *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025, pp. 1–12.

[62] Z. Li, C. Zhang, Z. Li, H. Howard-Jenkins, Z. Lv, C. Geng, J. Wu, R. Newcombe, J. Engel, and Z. Dong, "Art: Articulated reconstruction transformer," *arXiv preprint arXiv:2512.14671*, 2025.

[63] L. Le, J. Xie, W. Liang, H.-J. Wang, Y. Yang, Y. J. Ma, K. Vedder, A. Krishna, D. Jayaraman, and E. Eaton, "Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model," *arXiv preprint arXiv:2410.13882*, 2024.

[64] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Planar robot casting with real2sim2real self-supervised learning," *arXiv preprint arXiv:2111.04814*, 2021.

[65] R. Antonova, J. Yang, P. Sundaresan, D. Fox, F. Ramos, and J. Bohg, "A bayesian treatment of real-to-sim for deformable object manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5819–5826, 2022.

[66] X. Wang, L. Liu, Y. Cao, R. Wu, W. Qin, D. Wang, W. Sui, and Z. Su, "Embodiedgen: Towards a generative 3d world engine for embodied intelligence," *arXiv preprint arXiv:2506.10600*, 2025.

[67] M. N. Qureshi, S. Garg, F. Yandun, D. Held, G. Kantor, and A. Silwal, "Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 6502–6509.

[68] H. Xia, C.-H. Lin, H.-Y. Hsu, Q. Leboutet, K. Gao, M. Paulitsch, B. Ummenhofer, and S. Wang, "Holoscene: Simulation-ready interactive 3d worlds from a single video," 2025.

[69] M. Torne, A. Jain, J. Yuan, V. Macha, L. Ankile, A. Simeonov, P. Agrawal, and A. Gupta, "Robot learning with super-linear scaling," *arXiv preprint arXiv:2412.01770*, 2024.

[70] C. Gu, H. Kang, J. Lin, J. Wang, D. Wu, S. Xie, F. Huang, J. Ge, Z. Gong, L. Li *et al.*, "Igen: Scalable data generation for robot learning from open-world images," *arXiv preprint arXiv:2512.01773*, 2025.

[71] G. Chhablani, X. Ye, M. Z. Irshad, and Z. Kira, "Embodiedsplat: Personalized real-to-sim-to-real navigation with gaussian splats from a mobile device," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 25 431–25 441.

[72] A. Escontrela, J. Kerr, A. Allshire, J. Frey, R. Duan, C. Sferrazza, and P. Abbeel, "Gaussgym: An open-source real-to-sim framework for learning locomotion from pixels," *arXiv preprint arXiv:2510.15352*, 2025.

[73] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conf on Robot Learning*, 2018.

[74] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Robotics: Science and Systems*, Daegu, Republic of Korea, 2023.

[75] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in neural information processing systems*, 1989.

[76] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, 1999.

[77] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," *Proceedings 2002 IEEE Int'l Conf on Robotics and Automation*, vol. 2, 2002.

[78] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

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

[80] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE Int'l Conf on Robotics and Automation (ICRA)*, 2024.

[81] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics and Automation Magazine*, vol. 17, 2010.

[82] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[83] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," in *Forty-first Int'l Conf on Machine Learning*, 2023.

[84] Y. Tian, Y. Yang, Y. Xie, Z. Cai, X. Shi, N. Gao, H. Liu, X. Jiang, Z. Qiu, F. Yuan *et al.*, "Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy," *arXiv preprint arXiv:2511.16651*, 2025.

[85] A. Myronenko and X. Song, "Point-set registration: Coherent point drift," 2009.

[86] X. Qiu, J. Yang, Y. Wang, Z. Chen, Y. Wang, T.-H. Wang, Z. Xian, and C. Gan, "Articulate anymesh: Open-vocabulary 3d articulated objects modeling," *arXiv preprint arXiv:2502.02590*, 2025.

[87] X. Wei, M. Liu, Z. Ling, and H. Su, "Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search," 2022.

[88] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[89] C. Xu, Y. Chen, H. Wang, S.-C. Zhu, Y. Zhu, and S. Huang, "Partafford: Part-level affordance discovery from 3d objects," *arXiv preprint arXiv:2202.13519*, 2022.

[90] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1778–1787.

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

[92] I. Robotics, "Yam robot arm," 2025. [Online]. Available: https://i2rt.com/collections/yam-arm

[93] Q. Zhang, Z. Liu, H. Fan, G. Liu, B. Zeng, and S. Liu, "Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation," 2024.

[94] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "$\pi 0.5$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[95] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[96] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik, "Sam 3d: 3dfy anything in images," 2025. [Online]. Available: https://arxiv.org/abs/2511.16624

[97] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.

[98] Z. Wang, S. Chen, L. Yang, J. Wang, Z. Zhang, H. Zhao, and Z. Zhao, "Depth anything with any prior," 2025. [Online]. Available: https://arxiv.org/abs/2505.10565

[99] T. Hunyuan3D, S. Yang, M. Yang, Y. Feng, X. Huang, S. Zhang, Z. He, D. Luo, H. Liu, Y. Zhao, Q. Lin, Z. Lai, X. Yang, H. Shi, Z. Zhao, B. Zhang, H. Yan, L. Wang, S. Liu, J. Zhang, M. Chen, L. Dong, Y. Jia, Y. Cai, J. Yu, Y. Tang, D. Guo, J. Yu, H. Zhang, Z. Ye, P. He, R. Wu, S. Wei, C. Zhang, Y. Tan, Y. Sun, L. Niu, S. Huang, B. Zheng, S. Liu, S. Chen, X. Yuan, X. Yang, K. Liu, J. Zhu, P. Chen, T. Liu, D. Wang, Y. Liu, Linus, J. Jiang, J. Huang, and C. Guo, "Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material," 2025. [Online]. Available: https://arxiv.org/abs/2506.15442

[100] C. Ma, Y. Li, X. Yan, J. Xu, Y. Yang, C. Wang, Z. Zhao, Y. Guo, Z. Chen, and C. Guo, "P3-sam: Native 3d part segmentation," *arXiv preprint arXiv:2509.06784*, 2025.

[101] G. Tang, W. Zhao, L. Ford, D. Benhaim, and P. Zhang, "Segment any mesh," *arXiv preprint arXiv:2408.13679*, 2024.

[102] M. Liu, M. A. Uy, D. Xiang, H. Su, S. Fidler, N. Sharp, and J. Gao, "Partfield: Learning 3d feature fields for part segmentation and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 9704–9715.

APPENDIX

A. Task Rubric

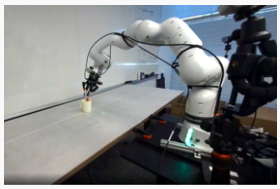
In this section, we provide the scoring rubric for each task, along with the language instruction provided to the VLAs. Each subtask is weighted equally and for each rollout, the number of completed subtasks is divided by the number of subtasks to give a normalized score between 0 and 1.

Task: Cup in Bowl

Instruction: Pick up the cup, and put the cup inside the bowl.
Rubric: 1. Touch cup 2. Pick cup 3. Place in bowl

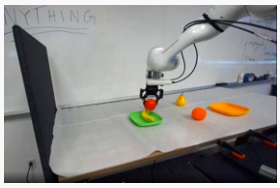
Task: Marker in Cup

Instruction: Pick up the marker, and put the marker in the cup.
Rubric: 1. Touch marker 2. Pick up marker 3. Place marker in cup



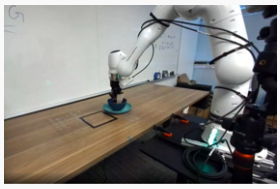
Task: Serve Fruits

Instruction: Put both the banana and the apple on the green plate.
Rubric: 1. Pick up banana 2. Pick up apple 3. Place banana on green plate 4. Place apple on green plate



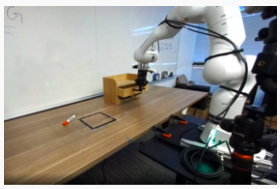
Task: Stack Dishware

Instruction: Put the bowl on the plate, then put the cup in the bowl.
Rubric: 1. Pick up bowl 2. Place bowl on plate 3. Pick up cup 4. Place cup in bowl



Task: Put Away Marker

Instruction: Open drawer, pick up marker, place marker in drawer, then close drawer.
Rubric: 1. Open cabinet drawer 2. Pick up marker 3. Place marker in drawer 4. Close drawer

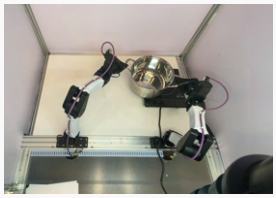


Task: Throw Away Trash

Instruction: Open the trash can, pick up the cup, put the cup into the trash can, and then close the trash can.
Rubric: 1. Open trash can 2. Pick up cup 3. Place cup inside trash can 4. Close trash can

Task: Pot On Stove

Instruction: Put the pot on the stove
Rubric: 1. Left arm grasps pot handle 2. Right arm grasps pot handle 3. Pot is lifted 4. Pot is placed on Stove



B. Policy Training Details

When reconstructing our scenes in simulation, we utilize SIMFOUNDRY to reconstruct the foreground (task relevant) objects, and separate methods for reconstructing the background. For the DROID setup, we utilize Scaniverse to generate a textured 3D visual mesh, and for the YAM setup, we directly model the the workcell geometry as a composition of textured 3D CAD files received from the supplier of our real YAM robots. In both cases, we manually align the background with the foreground objects. When running MimicGen on our source demos to synthetically generate demos, we apply additional forms of domain randomization across all generated demos, including material randomization, camera pose randomization, and (specifically in the DROID setup) table height randomization.

C. 3D Reconstruction Evaluation Details

When reconstructing our real world scenes in simulation using SIMFOUNDRY, we need to measure ground truth poses in order to evaluate our 3D reconstruction metrics. We achieve this by measuring quasi-ground truth object poses using FoundationPose. Because each scene has occlusion, we sequentially stage each benchmark scene by placing the object furthest in the background, recording its (fully unoccluded) inferred quasi-ground truth pose from FoundationPose, and repeating the process for each subsequent object until all objects are placed in the staged scene. When running our evaluations, only the final staged (occluded) scene image is used as input for all methods.

D. More Real2Sim Reconstruction Results

We showcase additional scene reconstruction results in Table VII. These additional scenes showcase SIMFOUNDRY's applicability across a broad range of realistic everyday scenes and objects.

E. Human Intervention Details

In addition to being fully automated, we provide a unified GUI providing readily accessible touchpoints with our

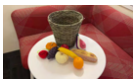
N Objects	Wall Clock Time (s)	Time per Object (s)	Original Image	Reconstructed Twin	Cousins Image
11	4027.96	366.18			
10	3060.61	306.061			
10	3522.71	352.271			

TABLE VII: **Additional real-to-sim reconstruction results. We show additional real-world scene input images, alongside SIMFOUNDRY's generated output and a sampled object variation instance as well as corresponding wall-clock time measurements for running our pipeline.**

system to allow for human operators to easy tune our pipeline's intermediate outputs. For example, during the scene decomposition process, a human operator can intervene and enforce specific constraints on individual objects being extracted, and can quickly tweak the generated pose and scale of meshes generated during the generation process. Likewise, during the scene variation process, the user can override and directly specify their own desired modifications to be made to specific objects, and directly iterate based on the resulting altered object image and mesh. By constructing our pipeline in a modular way, we both benefit from further model improvements released in the community and enable end-users to adjust final scene outputs according to their preferences.

- Show clear flow graph / pictures of I/O during the extraction process, especially step 5 decompose scene (e.g.: show all our intermediate outputs and how they are sequentially created)
- Qualitative examples of original object + image, modified cousin object meshes + images

F. VLM Details

SIMFOUNDRY is intended to be modular, and supports multiple VLMs that can be changed during execution. Below, we show the models our pipeline currently supports for each type of foundation model V_*:

- $V_{im2depth}$: If the input is a single image or video, we utilize DepthAnything3 [48]; if the input is a stereo image pair, we utilize FoundationStereo [46].
- V_{seg}^{image} : We utilize SAM3 [41].
- V_{scene} : We utilize Gemini-Pro-3, though any Gemini or other general purpose VLM can be used by our pipeline.
- V_{image} : We utilize Gemini-Pro-3-Image-Preview, though any Gemini or other general purpose image-editing VLM can be used by our pipeline.
- $V_{inpaint}^{depth}$: We utilize PriorDepthAnything [98].
- V_{mesh}: We utilize either Hunyuan2.1 [99] or TRELLIS.2 [40].
- $V_{articulation}$: We utilize Gemini-Pro-3, though any Gemini or other general purpose VLM can be used by our pipeline.
- V_{seg}^{mesh} : We utilize mainly P3-SAM [100], although our pipeline also supports Segment Any Mesh [101] and Partfield [102].

G. Articulated Object Generation

In this section we detail our articulated object generation pipeline, which extends prior methods such as Articulate Anymesh [86] and Articulate Anything [63].

a) Segmentation: We first render views of the object from multiple angles, pass these into a VLM $V_{articulation}$, and prompt $V_{articulation}$ to list the different parts of the object that can be articulated and the types of joints. For example, this would be drawers (prismatic) for a cabinet or door (revolute) for a microwave. We then segment the mesh of the object with a mesh segmentation model V_{seg}^{mesh} , such as P3-SAM [100] or Segment Any Mesh [101] which assigns a label to every face in the model. Most existing methods only assign labels to external surfaces, but since meshes generated by TRELLIS.2 [40] can have internal structures, we propagate these labels to unlabeled mesh faces using a graph label propagation algorithm. This usually results in an over-segmented model, so we once again render the object with the different segments labeled, and prompt $V_{articulation}$ to assign each segment to the parts from the previous step. This segmentation and assignment can be refined by the user via a GUI. The relevant segments for each part are then merged into separate meshes.

b) Joint Generation: We adapt the actor-critic algorithm and API from Articulate Anything [63] to generate the joint parameters. We prompt $V_{articulation}$ to predict the joint axes and placements of each part, by providing a python API which can generate URDFs. The API allows $V_{articulation}$ to place joints relative to parts (for example, a revolute joint can be placed along the left edge of a door), which helps ground $V_{articulation}$ and simplify its task. $V_{articulation}$ generates code which calls this API, and the result is compiled into a URDF. We then move the joints of the object according to this URDF in a simulator, and render a video of this movement. The video is judged by another VLM, which is asked to rate the accuracy and realism of the movement, and provide feedback for improvement if necessary. $V_{articulation}$ is prompted to improve its prediction, by incorporating this feedback, and this process continues until the critic gives a score above a threshold.

c) Physical Parameters: Finally, we prompt $V_{articulation}$ to generate the physical parameters such as link mass, joint friction, and damping. We provide $V_{articulation}$ with the calculated volume of each part and the entire object to aid in this.