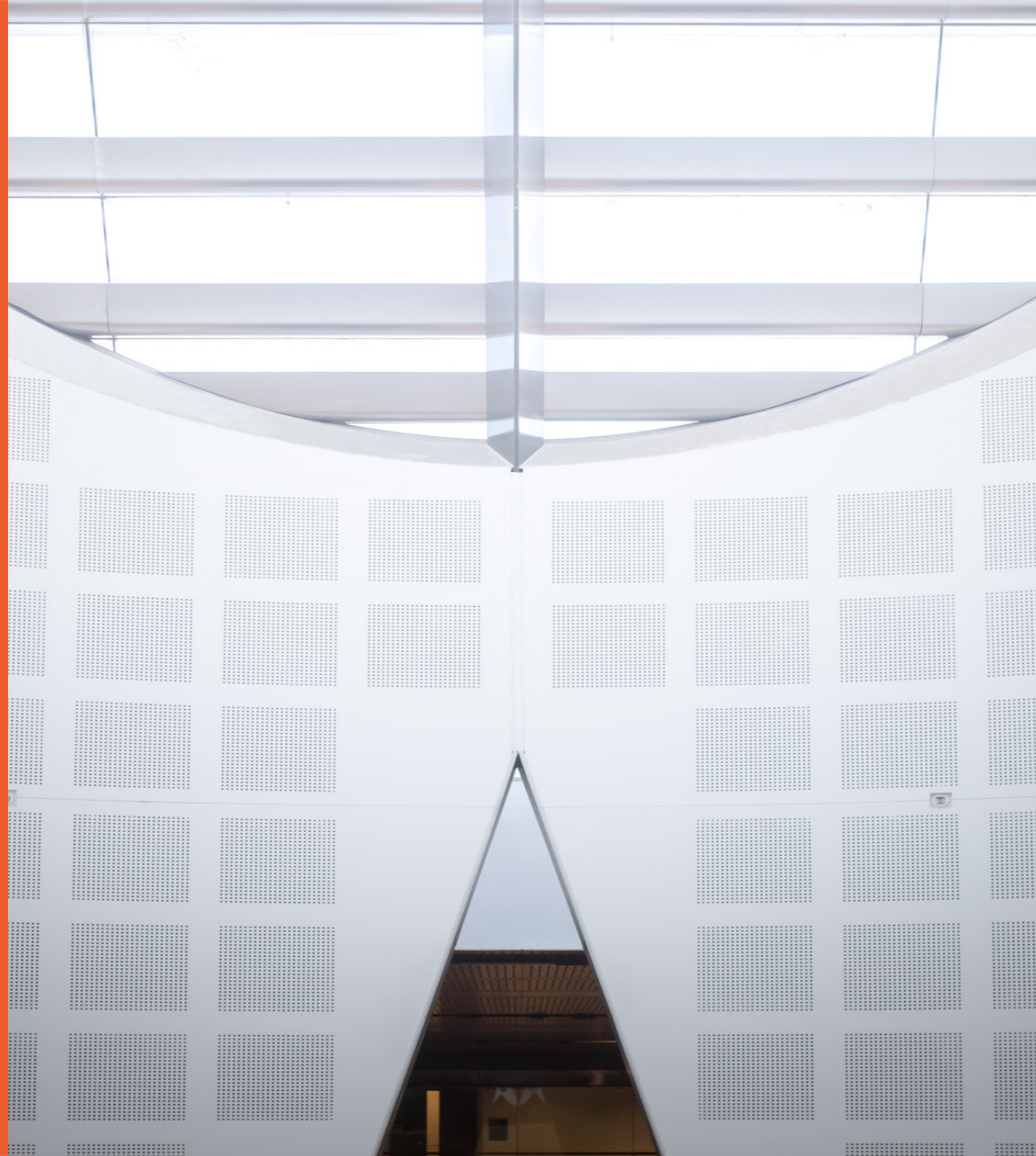
Seminar 2:
LLM and VLM for Robotics

Xiyu Wang

School of Computer Science

THE UNIVERSITY OF SYDNEY



RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

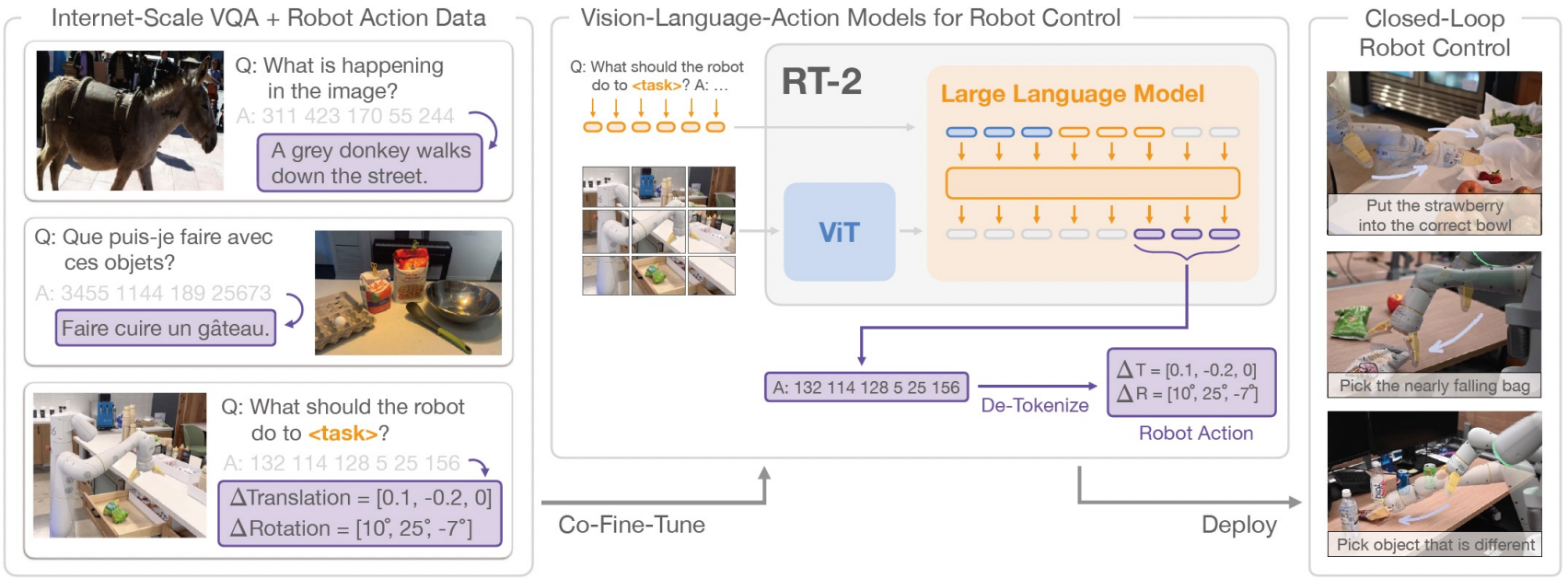


Figure 1 | RT-2 overview: we represent robot actions as another language, which can be cast into text tokens and trained together with Internet-scale vision-language datasets. During inference, the text tokens are de-tokenized into robot actions, enabling closed loop control. This allows us to leverage the backbone and pretraining of vision-language models in learning robotic policies, transferring some of their generalization, semantic understanding, and reasoning to robotic control. We demonstrate examples of RT-2 execution on the project website: robotics-transformer2.github.io.

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

1. **For Training and Prediction**: Use the LLM and VLM to generate videos for training and prediction purposes.

2. **For Simulation Data**: Leverage the LLM and VLM to create simulation data for training.

Gen2Act

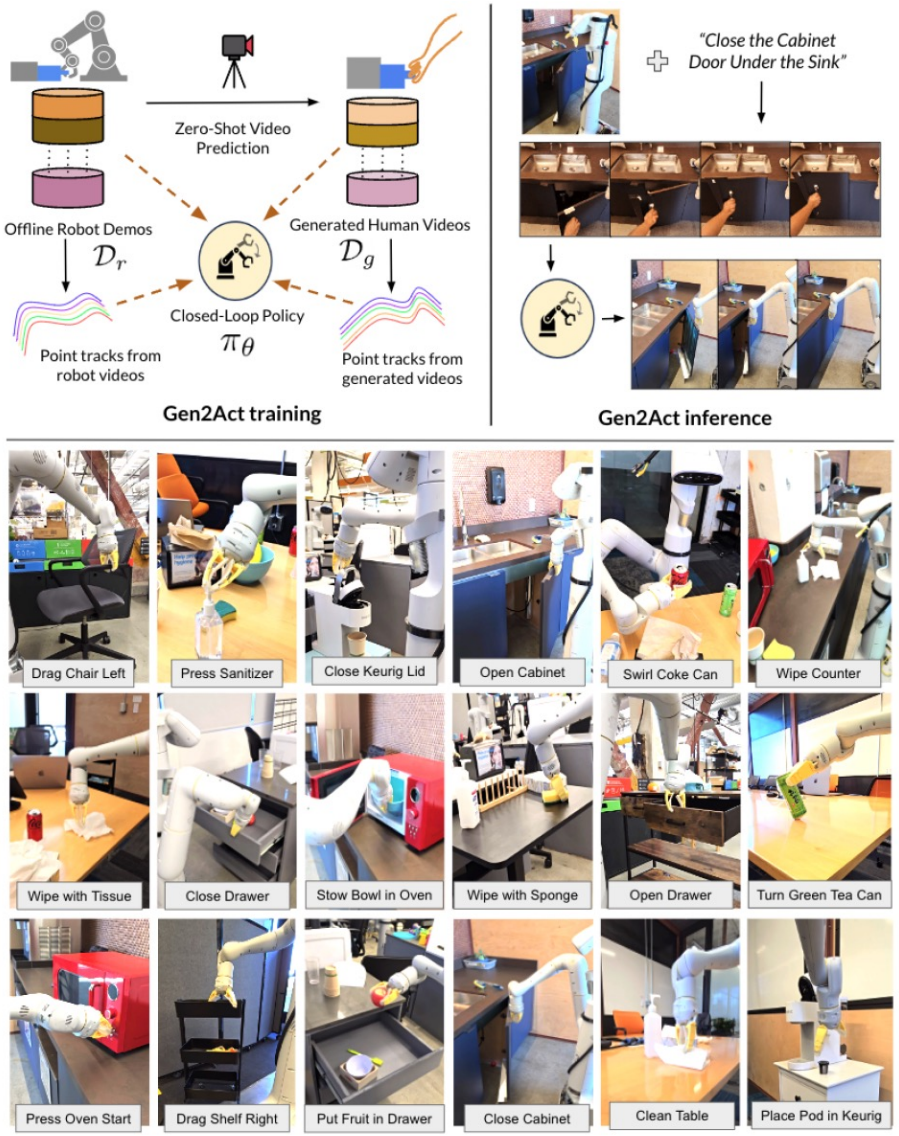


Fig. 1: *Gen2Act* learns to generate a human video followed by robot policy execution conditioned on the generated video. This enables diverse real-world manipulation in unseen scenarios.

Problem Solved: When facing new tasks, robotic operation strategies often struggle with unfamiliar object types and new actions. Given the high cost of robot data collection, generalizing operation strategies is a key challenge.

Proposed Solution: The Gen2Act method is proposed, which predicts motion information from internet data to generate human videos, and combines robotic strategies with these generated videos. This approach guides robotic strategies to perform new tasks through zero-sample human video generation, avoiding extensive robot data collection.

Technologies Applied: Zero-sample human video generation under language conditions. Use a pretrained video generation model to directly produce human videos without the need for fine-tuning the model. Utilize a small amount of robot interaction data to train the strategy model, and combine it with the generated videos to execute tasks.

Gen2Act

Fig. 3: Visualization of zero-shot video generation for different tasks. The blue frame and the language description are input to the video generation model of *Gen2Act* and the black frames show sub-sampled frames of the generated video. These results demonstrate the applicability of off-the-shelf video generation models for image+text conditioned video generation that preserves the scene and performs the desired manipulation task.

Gen2Act

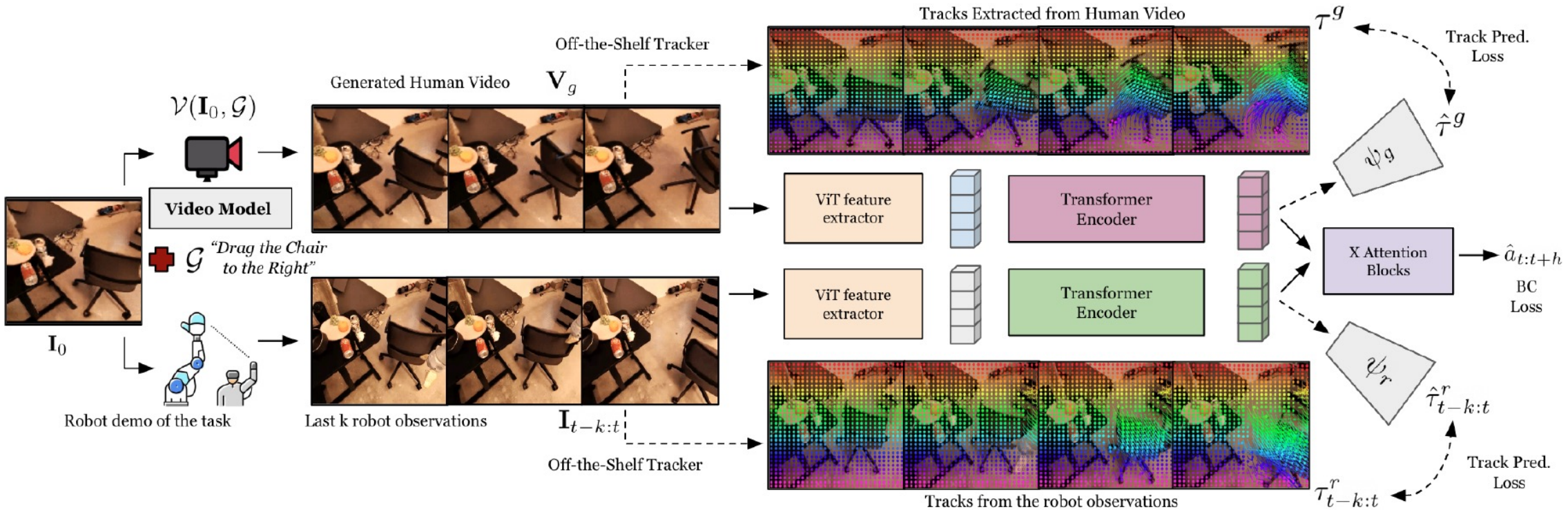


Fig. 2: Architecture of the translation model of *Gen2Act* (closed-loop policy π_θ). Given an image of a scene $\mathbf{I}_0$ and a language-goal description of the task $\mathcal{G}$, we generate a human video $\mathbf{V}_g$ with a pre-trained video generation model $\mathcal{V}(\mathbf{I}_0, \mathcal{G})$. During training of the policy, we incorporate track prediction from the policy latents as an auxiliary loss in addition to a behavior cloning loss. Dotted pathways show training-specific computations. During inference, we do not require track prediction and only use the video model $\mathcal{V}$ in conjunction with the policy $\pi_\theta(\mathbf{I}_{t-k:t}, \mathbf{V}_g)$.

Gen2Act:

- Generated human video
- Visual feature extraction
- Point trajectory prediction

Gen2Act

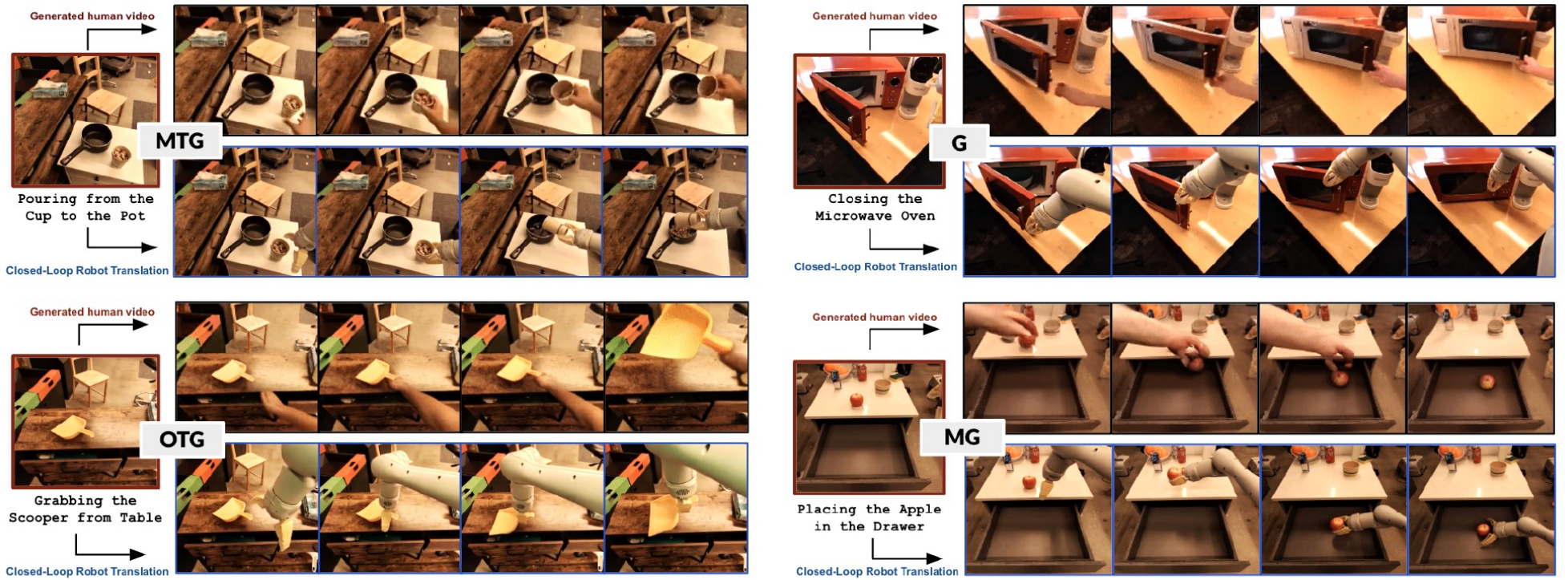


Fig. 4: Visualization of the closed-loop policy rollouts (bottom row) conditioned on the generated human videos (top row) for four tasks. The red frame and the language description are input to the video generation model of *Gen2Act* . The black frames show sub-sampled frames of the generated video, and the blue frames show robot executions conditioned on the generated video.

Deployment:

- A large-scale language model (such as Gemini) to obtain language descriptions of different tasks.
- Generate videos sequentially rather than generating all videos from the initial image

Gen2Act

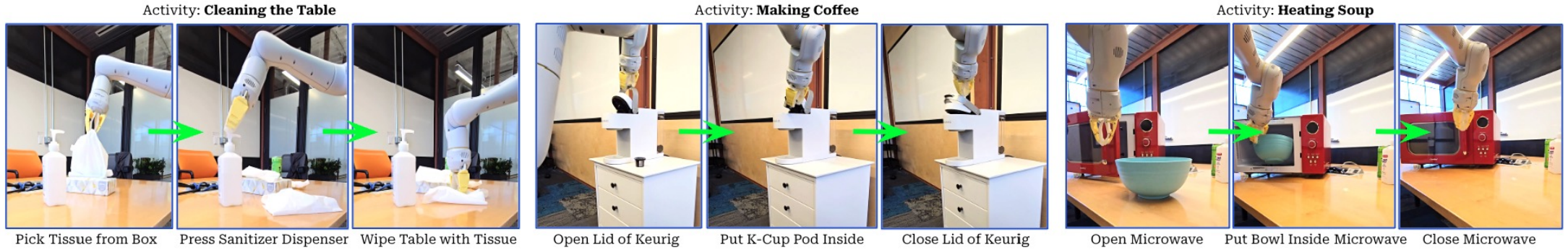


Fig. 5: Robot executions for a sequence of tasks. The last frame of the previous execution serves as the conditioning frame for next stage video generation.

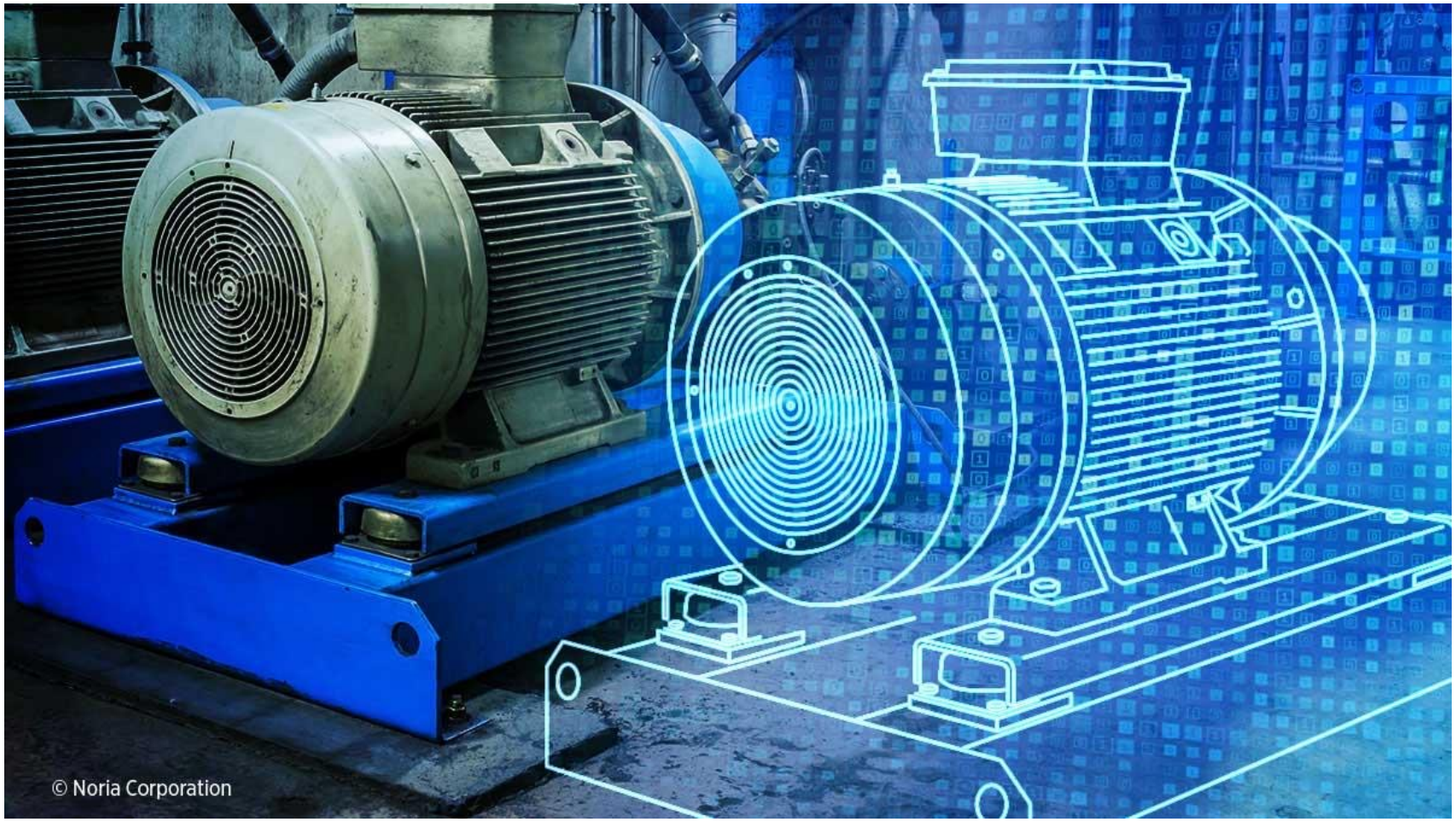
TABLE II: Comparison of success rates for long-horizon activities via chaining of different tasks. We first obtain sub-tasks for activities with an off-the-shelf LLM and then rollout *Gen2Act* in sequence for the different intermediate tasks.

Activity	Stages (from Gemini)	Success % Stage 1, Stage 2, Stage 3
Stowing Apple	1. Open the Drawer 2. Place Apple in Drawer 3. Close the Drawer	80, 60, 60
Making Coffee	1. Open the Lid 2. Place K-Cup Pod inside 3. Close the Lid	40, 20, 20
Cleaning Table	1. Pick Tissues from Box 2. Press the Sanitizer Dispenser 3. Wipe the Table with Tissues	60, 40, 40
Heating Soup	1. Open the Microwave 2. Put Bowl inside Microwave 3. Close the Microwave	40, 20, 20

TABLE III: Analysis of co-training with an additional dataset of diverse tele-operated robot demonstrations ($\sim$ 400 trajectories).

Co-Training	Mild (MG)	Standard (G)	Obj. Type (OTG)	Motion. Type (MTG)	Avg.
Gen2Act (w/o co-train)	83	67	58	30	60
Gen2Act (w/ co-train)	85	75	62	35	64

What is Digital Twin

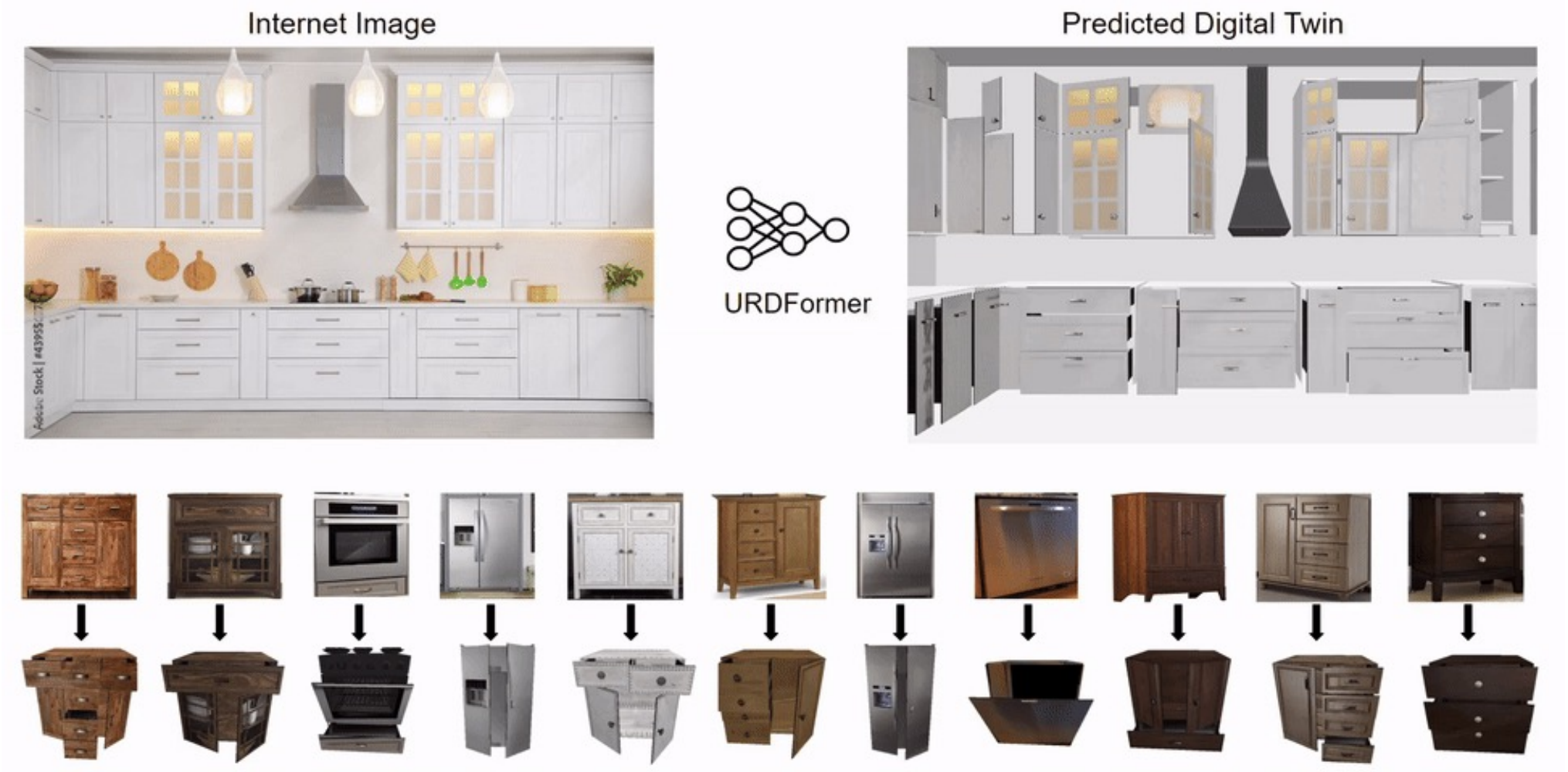


A digital twin is a virtual model of a physical object. It spans the object's lifecycle and uses real-time data sent from sensors on the object to simulate the behaviour and monitor operations.

ReplicaCAD

Figure 2: Left: The original Replica scene. Right: the artist recreated scene ReplicaCAD. All objects (furniture, mugs) including articulated ones (drawers, fridge) in ReplicaCAD are fully physically simulated and interactive.

URDFormer



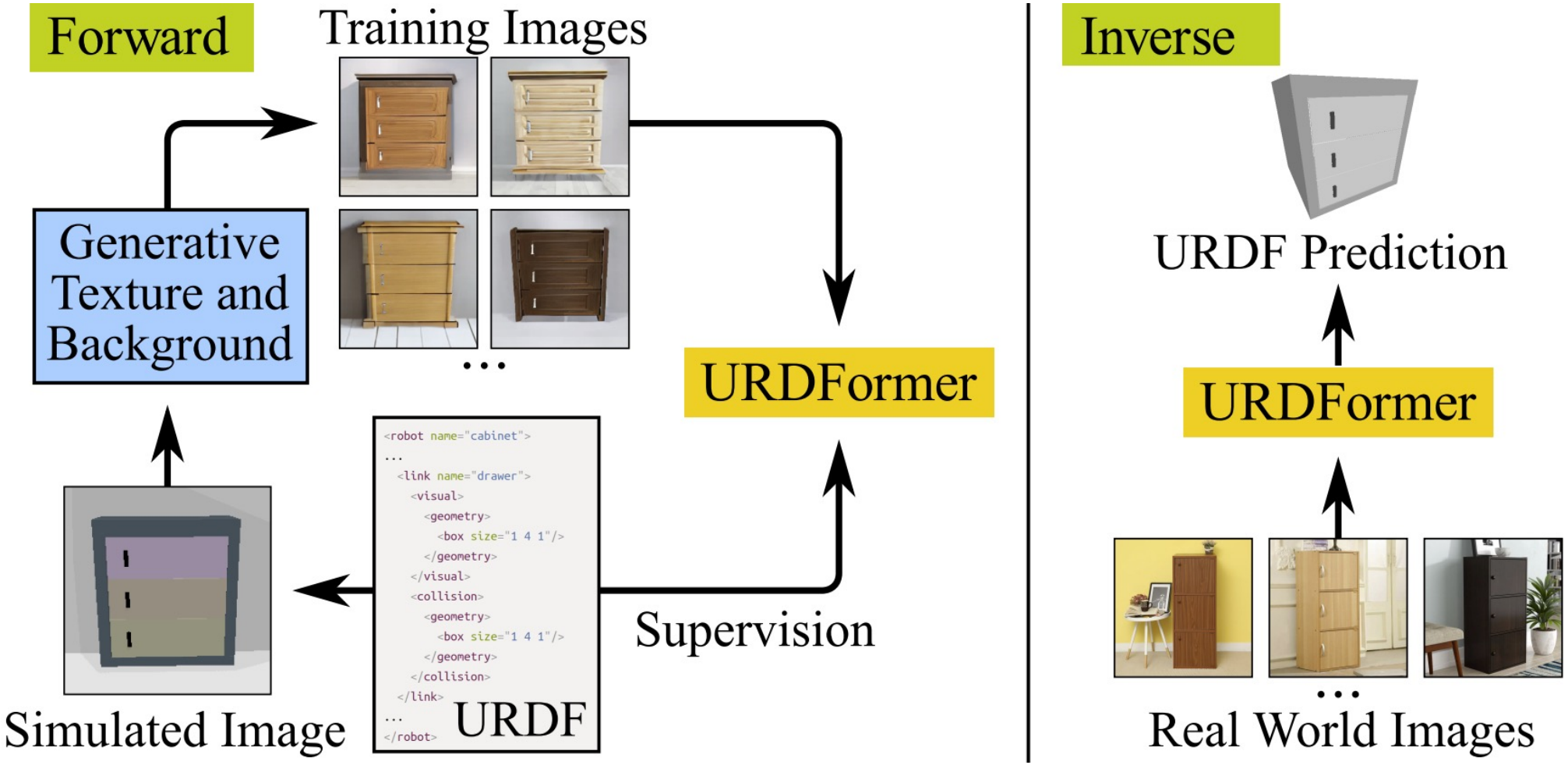
Their goal is to train a network to predict a scene that matches the kinematic structure of the image, and it's fully interactive to train a robot for different tasks.

URDFormer


Large Simulation Data
Diffusion Models
Generated Realistic Images
Real-World Data

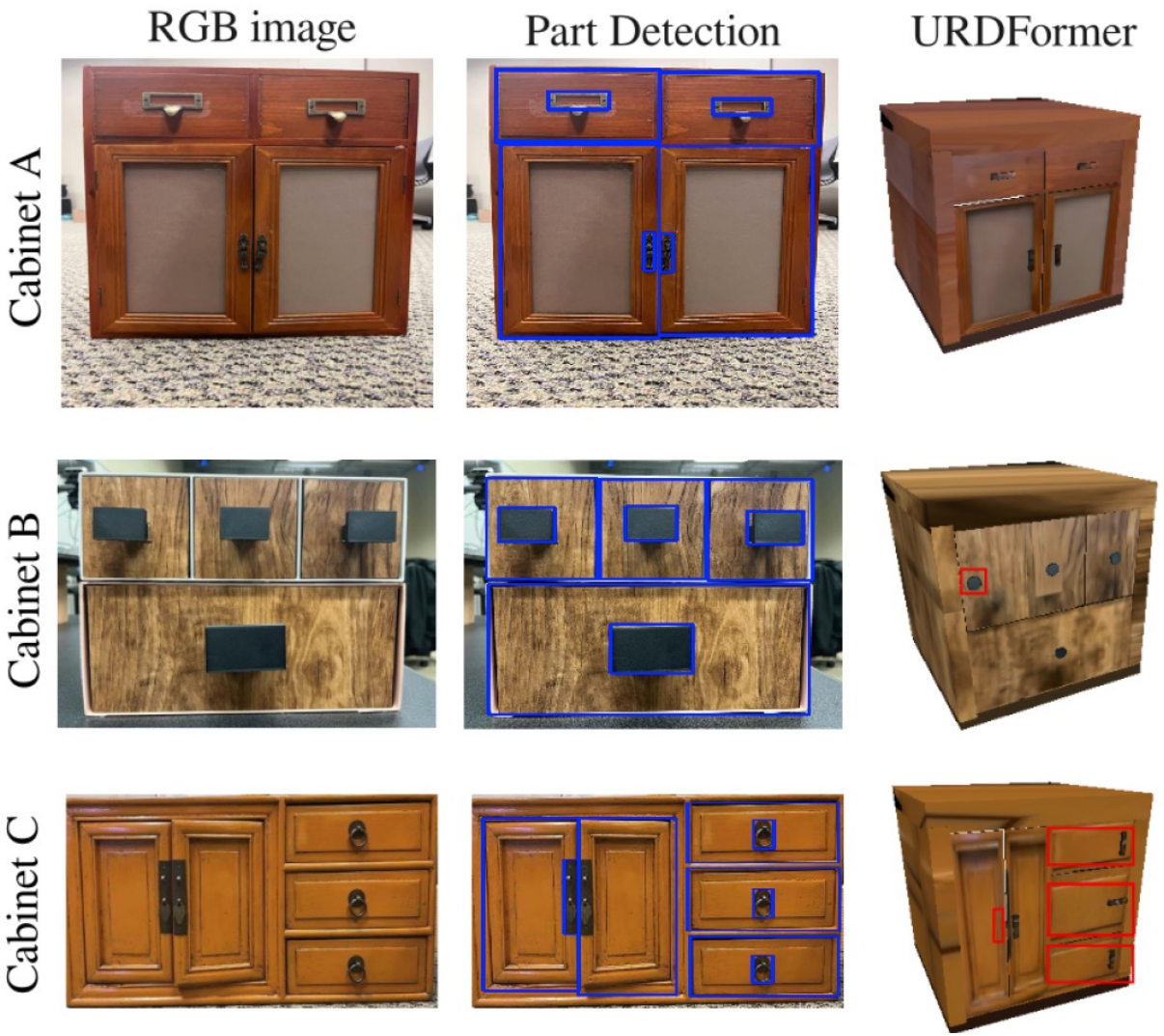

Simulation
Generated Paired Images
Object-level
Scene-level

URDFormer

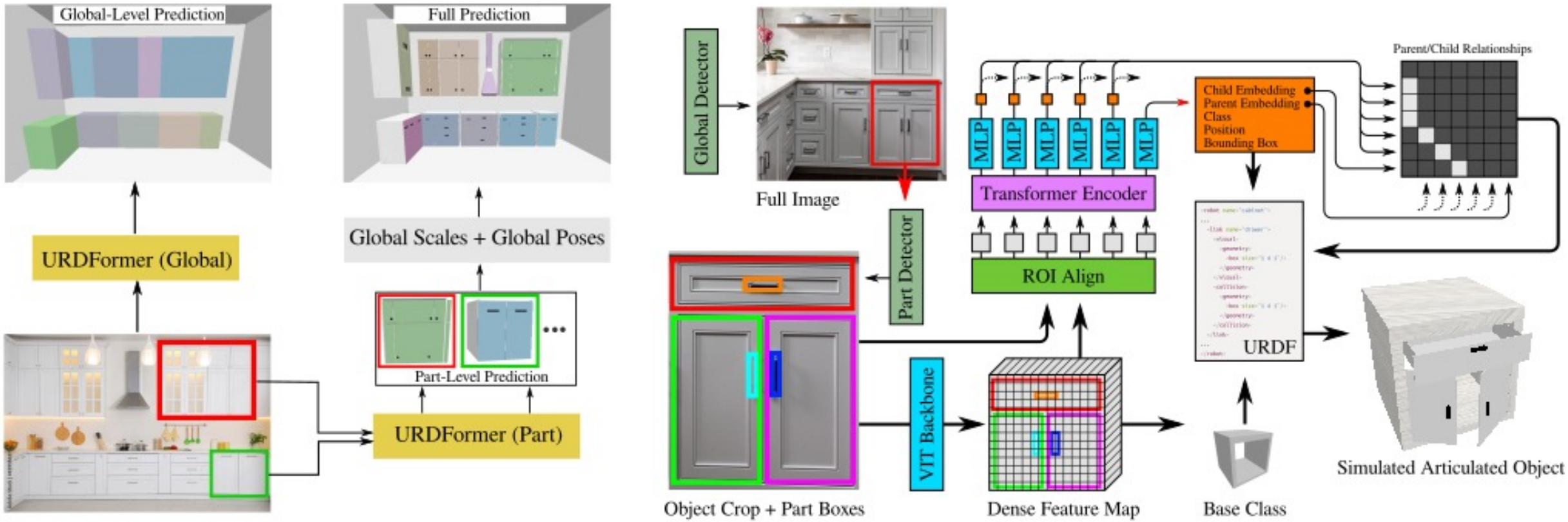


The URDFormer is trained on a large paired dataset of simulation assets and realistic renderings (forward). During inference, this process is inverted and it predicts the unified robot description format (URDF) from a real image (inverse).

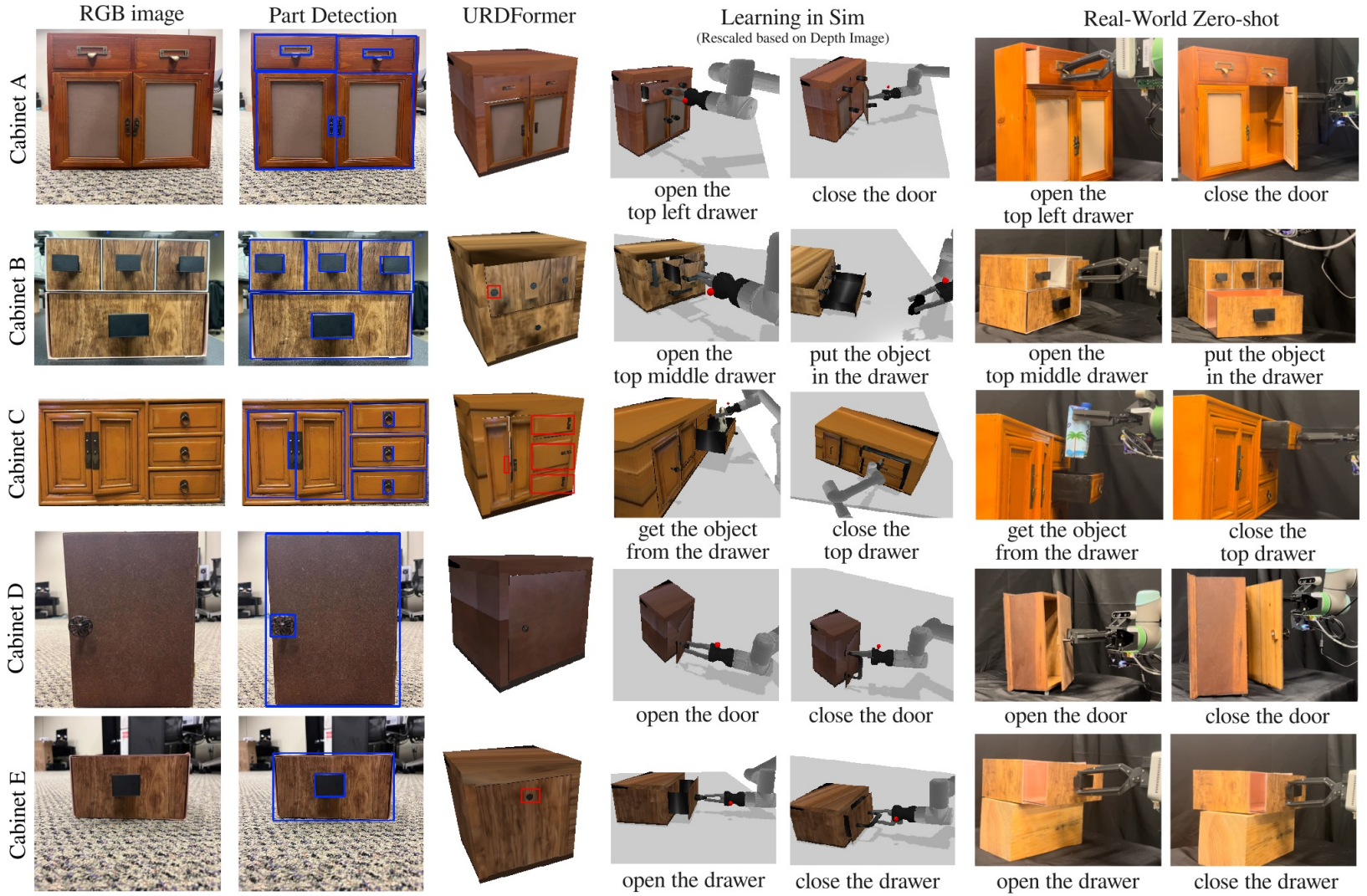
URDFormer


RGB image
Part Detection
URDFormer
Cabinet A
Cabinet B
Cabinet C

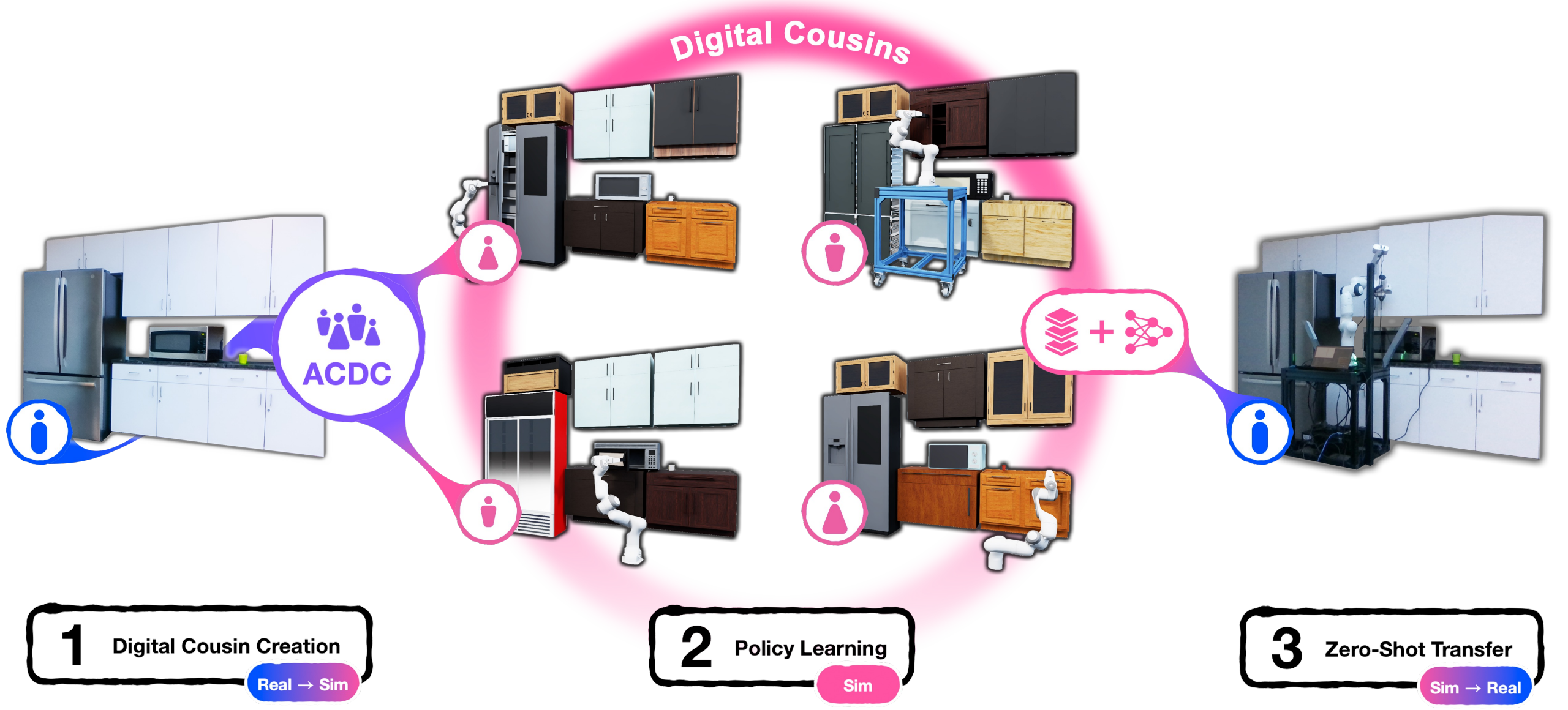
URDFormer


Global-Level Prediction
Full Prediction
URDFormer (Global)
Global Scales + Global Poses
Part-Level Prediction
URDFormer (Part)
Global Detector
Full Image
Part Detector
Object Crop + Part Boxes
VIT Backbone
Dense Feature Map
MLP
Transformer Encoder
ROI Align
Child Embedding
Parent Embedding
Class
Position
Bounding Box
Parent/Child Relationships
URDF
Base Class
Simulated Articulated Object

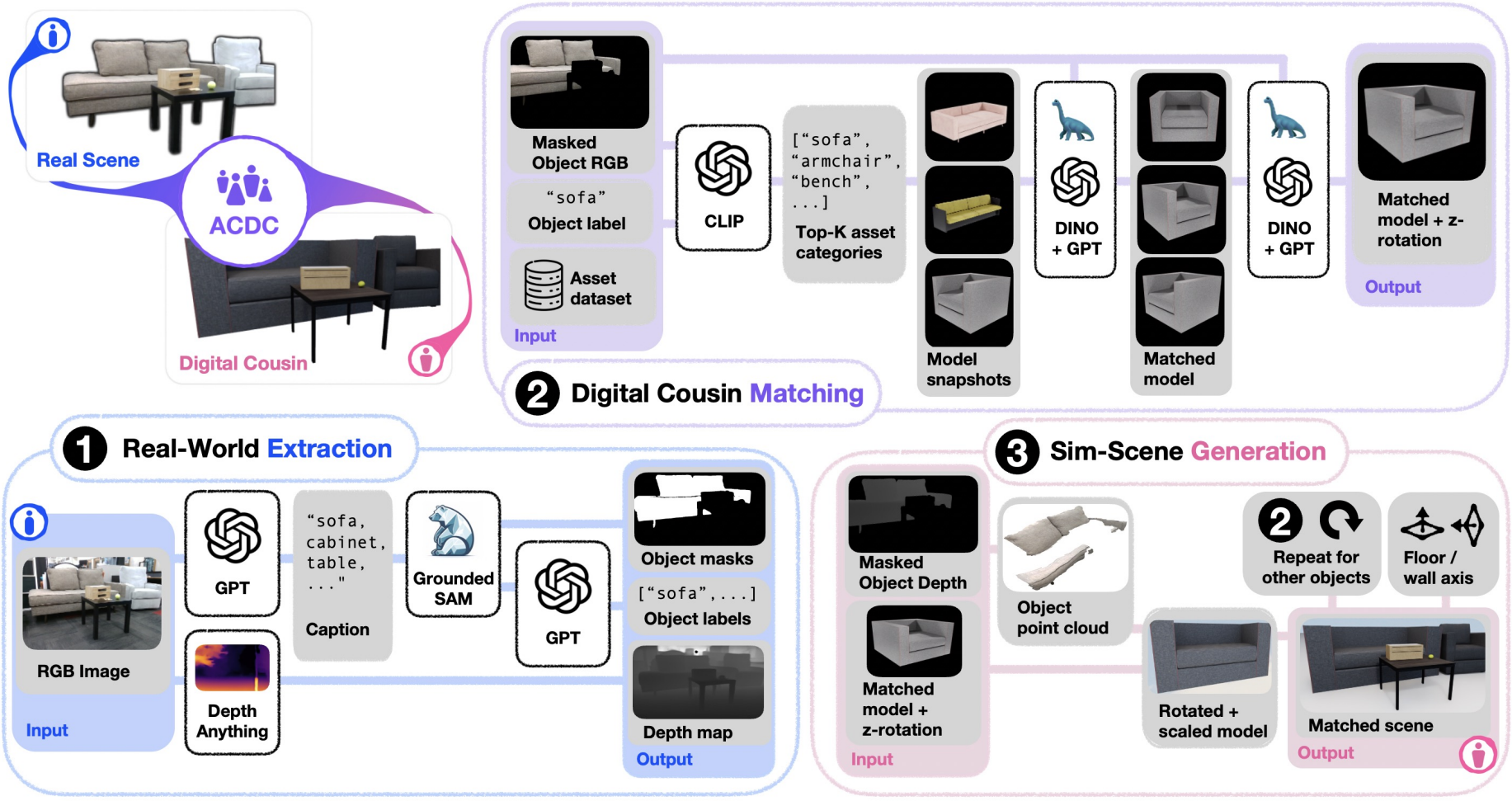
URDFormer


RGB image
Part Detection
URDFormer
Learning in Sim
(Rescaled based on Depth Image)
Real-World Zero-shot
Cabinet A
Cabinet B
Cabinet C
Cabinet D
Cabinet E
open the top left drawer
close the door
open the top left drawer
close the door
open the top middle drawer
put the object in the drawer
open the top middle drawer
put the object in the drawer
get the object from the drawer
close the top drawer
get the object from the drawer
close the top drawer
open the door
close the door
open the door
close the door
open the drawer
close the drawer
open the drawer
close the drawer

ACDC


Digital Cousins
ACDC
1 Digital Cousin Creation
Real → Sim
2 Policy Learning
Sim
3 Zero-Shot Transfer
Sim → Real

ACDC



ACDC Pipeline. ACDC is composed of three sequential steps.

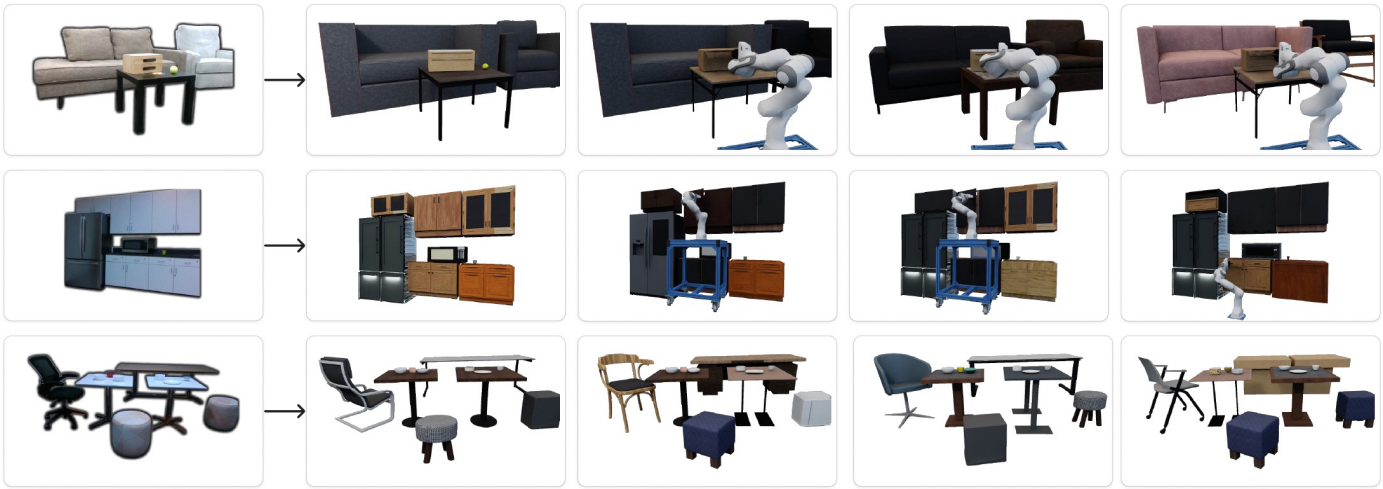
(1) First, relevant per-object information is extracted the input RGB image.

(2) Next, we use this information with an asset dataset to match digital cousins to each detected input object.

(3) Finally, we post-process the chosen digital cousins and generate a fully-interactive simulated scene.

ACDC

Input Scene	ACDC Output	Scale (m)	Cat.	Mod.	$\mathcal{L}_2$ Dist. (cm) ↓	Ori. Diff. (rad) ↓	Bbox IoU ↑	Cen. IoU ↑
		3.42	6/6	6/6	4.15 ± 2.04	0.10 ± 0.14	0.64 ± 0.23	0.73 ± 0.22
		4.17	8/8	8/8	7.65 ± 5.62	0.05 ± 0.00	0.66 ± 0.21	0.74 ± 0.16
		6.89	10/10	10/10	4.77 ± 3.38	0.03 ± 0.01	0.74 ± 0.20	0.77 ± 0.19
		10.23	15/15	15/15	15.67 ± 8.86	0.12 ± 0.11	0.59 ± 0.14	0.72 ± 0.14



Input RGB Image

Reconstructed Scenes

Policy	Sim Success	Real Success
Twin	100%	25%
Twin + ↑ DR	70%	55%
Twin + Cousin	92%	95%
Cousin	94%	90%

Figure 5: **Zero-shot real-world evaluation of digital cousin policy vs. digital twin baselines.** Task is **Door Opening** on an IKEA cabinet. Metric is success rate: sim/real results averaged over 50/20 trials. Twin + ↑DR is trained using increased domain (pose, scale) randomization, and Twin + Cousin is trained on both twin and cousin data.