LUCID: Learning Embodiment-Agnostic Intent Models from Unstructured Human Videos for Scalable Dexterous Robot Skill Acquisition

Harsh Gupta[†]
University of Illinois Urbana-Champaign
hgupt3@illinois.edu

Guanya Shi[*]
Carnegie Mellon University
guanyas@andrew.cmu.edu

Wenzhen Yuan[*]
University of Illinois Urbana-Champaign
yuanwz@illinois.edu

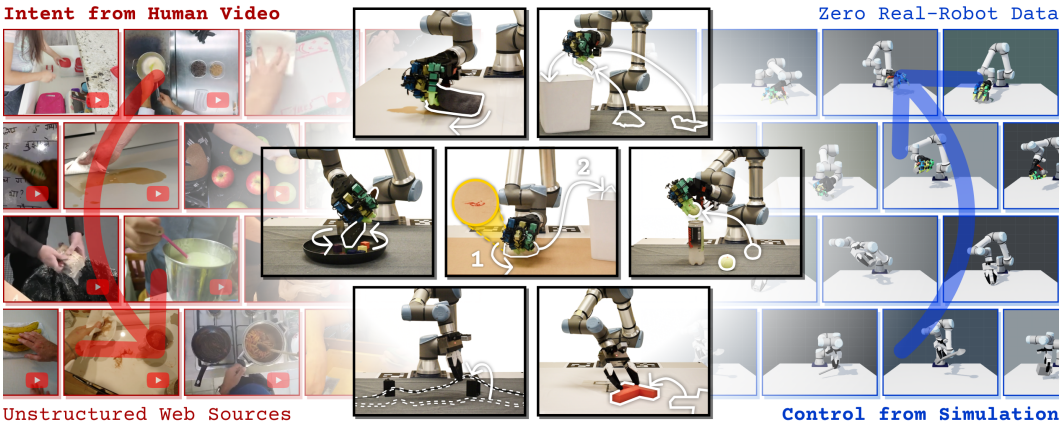

Figure 1: LUCID. We learn a manipulation intent model from human video (left) and a robot controller policy from simulation (right), and pair them in real-world deployment on a dexterous hand and a parallel-jaw gripper (center).

Abstract: The most widely-adopted robot learning pipelines today learn skills from robot demonstrations or structured human data, which are expensive to collect and tied to specific embodiments. In contrast, unstructured human videos provide a scalable alternative. They contain diverse manipulation demonstrations across objects, scenes, and strategies, but are not directly connected to robot action. We propose LUCID, a two-stage framework that learns task intent from unstructured human videos drawn from internet-scale datasets and learns robot control in massively-parallel simulation. The intent model predicts short-horizon intent (what should happen next in the scene) from the current observation in closed loop. An embodiment-specific sensorimotor policy converts this intent into robot actions. The intent interface is shared across controllers, so the same intent model can be applied to different embodiments, from our primary dexterous hand to a parallel-jaw gripper. We evaluate LUCID on five real-world manipulation tasks: stirring, wiping, and binning supervised by only internet video, with zero-shot transfer to novel scenes and object instances; and push-T and cable routing supervised by 1 hr each of self-collected smartphone video. Project page: https://lucid-robot.github.io/.

Keywords: Robot manipulation; Reinforcement learning for physical robot control; Learning from human videos; Sim-to-real transfer

[†]Corresponding author. [*]Equal advising.

1 Introduction

Robot manipulation policies are typically trained on action-observation data, sourced either from teleoperated robot demonstrations [1, 2] or from human demonstrations captured with structured rigs or interfaces (motion capture, multi-view, wearables, handheld grippers) [3, 4, 5, 6]. These pipelines require purpose-built collection infrastructure, remain tied to a particular robot or interface, and scale only with operator hours. Two alternative sources scale beyond these limits. First, unstructured human video (internet videos) is abundant and broad in objects and strategies, but actionless. Second, physical simulation produces action-labeled data at arbitrary scale, but each task needs hand-designed rewards, especially hard to define for high-level intent. We argue these two sources are naturally complementary and should be paired through an intent-control separation: human videos provide embodiment-agnostic intent [7, 8], while massive simulation provides a task-agnostic and robust sensorimotor policy [9, 10].

Prior attempts to draw on these sources each have a distinct shortcoming. Imitation policies trained on extracted human-video trajectories learn trajectory-level behavior rather than task-level intent, and don't generalize beyond the demonstrated scene [11, 12]. Open-loop planners that condition a pretrained video model on the initial scene cannot recover when execution diverges [7, 13]. Pretrained video models repurposed as policy backbones still need per-task and per-embodiment robot data [14, 15]. On the simulation side, generalist sensorimotor policies generalize across objects under massive randomization [16, 17, 9], but their inference-time reference still comes from outside the policy (motion capture, single-video extraction, or a video model run at task start).

We propose `LUCID`, which pairs intent (what should change in the scene) and control (how the robot achieves it) through two design choices. First, intent and control are decoupled: an intent model f_θ trained on unstructured human video predicts short-horizon object flow and a palm-pose reference from the current scene, while a sensorimotor policy π trained once in simulation realizes these references on a dexterous hand-arm system, and a parallel-jaw gripper variant demonstrates embodiment transfer. Second, intent is predicted in closed loop: at deploy, f_θ continually re-queries from the live scene rather than producing a one-shot plan, with no object mesh, motion capture, or per-embodiment adaptation required.

We evaluate `LUCID` against four claims: it (1) learns intent that transfers zero-shot to novel scenes, camera viewpoints, and object instances, (2) improves robustness via closed-loop intent prediction, (3) extends across robot embodiments with the same intent model, and (4) improves predictably with the amount of training video, both in intent loss and downstream task success. We test these on five real-world manipulation tasks spanning web-scale video (stirring, wiping, binning) and self-collected smartphone video (push-T, cable routing), on a dexterous robot hand [18] and a parallel-jaw gripper. Closed-loop `LUCID` achieves 73% average success on the web-supervised tasks vs 28% for an open-loop baseline, and the same intent model drives both embodiments with comparable success (63%) on the smartphone-collected tasks.

2 Related Work

2.1 Dexterous robot manipulation from human video

Trajectories from video. A body of work trains policies directly on trajectories derived from humans performing the task, using uninstrumented internet video reconstructed or edited into 3D hand and object trajectories [11, 19, 20, 21, 22, 12, 23, 24, 25, 26] or in-scene video captured with wearables or calibrated cameras [4, 27, 28, 29]. These policies are scoped to specific recorded trajectories and limited by reconstruction noise and the human-to-robot embodiment gap. `LUCID` uses human video only for intent supervision.

Plans from video. A second line uses pretrained video generators, often off-the-shelf, to produce a plan at the start of a rollout from the initial observation [7, 13, 30, 31, 8]. A separate executor realizes the plan, and the plan itself is generated once at $t = 0$ and not updated during execution.

This open-loop framing reflects the cost of video generation [31, 30]. LUCID instead issues new short-horizon references continually from the current observation.

Representations from video. A third line treats human video as a pretraining supplement for systems that still require robot teleoperation or real-world demonstrations, whether as a pretrained video model used as a runtime component [14, 15, 32, 33, 34] or as pretraining for a VLA or transformer backbone that is then co-trained or fine-tuned with robot demonstrations [35, 36, 37, 38, 39, 40, 41, 42]. Across these variants, robot action data remains the bottleneck. LUCID avoids teleoperation entirely, supervising intent from human video and control from simulation.

2.2 Sim-to-real RL for dexterous robot control

Reinforcement learning in simulation is commonly used to train dexterous robot policies against reference hand and object trajectories from motion capture or video [43, 44, 45, 46, 12, 23], with each policy tied to the reference it was trained against. Recent work has scaled this lineage into generalist sensorimotor policies that span many trajectories or object instances under aggressive domain randomization, generalizing across objects, geometries, and scene conditions [16, 17, 9, 47]. References at inference still come from outside the policy (MoCap, teleop, single-video poses, or mesh+goal pose). LUCID pairs a generalist sensorimotor policy with an intent model that produces references in closed loop.

2.3 Intent representations for robot policies

Human video reveals intent but not actions, so prior work uses intermediate representations to bridge the two. Object-centric flow is one such representation: point tracks or per-point trajectories extracted from human video or video generators [48, 49, 41, 50, 7, 30, 8, 13, 51, 52]. Flow is mesh-free, covers rigid, articulated, and deformable objects, and readily extracted from internet video. But it does not specify where the hand should be at contact, an underspecification that matters for multi-fingered manipulation. Hand-centric representations (hand poses, wrist trajectories, grasp priors) carry the complementary functional grasp that flow cannot capture [53, 25, 11, 28, 44, 45, 54]. Executors are typically lightweight (inverse kinematics, trajectory optimization, residuals on heuristic policies) [30, 8, 31], and are often the primary failure mode. LUCID pairs flow and palm-pose intent with a generalist sensorimotor policy.

3 Method

This section details LUCID's two learned components: the intent model f_θ (§3.1) and the sensorimotor policy π (§3.2), which communicate through a short-horizon reference $\mathcal{R}$ comprising object flow and a palm pose. §3.3 describes how they run in closed loop at deploy.

3.1 Intent Model

We define manipulation intent as a short-horizon prediction of object motion and rough palm pose, shared across embodiments; the joint-level commands that realize it are delegated to a separate sensorimotor policy. We predict short-horizon object and hand motion from a current RGB-D observation (Fig. 2). Let $\mathbf{I}_t$ denote a stack of F recent RGB-D frames ending at time t, and let $\tau = 0, 1, \ldots, T$ index the current step and T future steps. For query point n on the target object, $\mathbf{x}^{\text{trk}}_{n,\tau} \in \mathbb{R}^3$ is its 3D position at step τ, and $(\mathbf{p}^{\text{palm}}_\tau, \mathbf{R}^{\text{palm}}_\tau) \in SE(3)$ is the palm pose at step τ. Given $\mathbf{I}_t$ and the current-step values of these quantities, the intent model f_θ predicts the future-step values, which stack into the short-horizon reference $\mathcal{R}$ consumed by the sensorimotor policy (§3.2):

$$f_\theta\big(\mathbf{I}_t,\ \{\mathbf{x}^{\text{trk}}_{n,0}\}_{n=1}^{N},\ (\mathbf{p}^{\text{palm}}_0, \mathbf{R}^{\text{palm}}_0)\big)\ =\ \mathcal{R}\ =\ \big(\{\mathbf{x}^{\text{trk}}_{n,\tau}\}_{n=1,\tau=1}^{N,T},\ \{(\mathbf{p}^{\text{palm}}_\tau, \mathbf{R}^{\text{palm}}_\tau)\}_{\tau=1}^{T}\big).$$

Architecture. The intent model f_θ adapts CoTracker3 [55] as a point-token transformer for short-horizon prediction. We make three changes. (1) We condition the transformer on frozen DINOv3 [56] patch tokens and fuse depth through a residual adapter added to the patch-token space. (2) We predict forward in time from $\mathbf{I}_t$, whereas standard point trackers estimate tracks over already-observed frames. (3) We attach a single palm-pose token alongside the N object tokens, so object flow and palm pose are produced jointly. See App. A.2 for the full configuration.

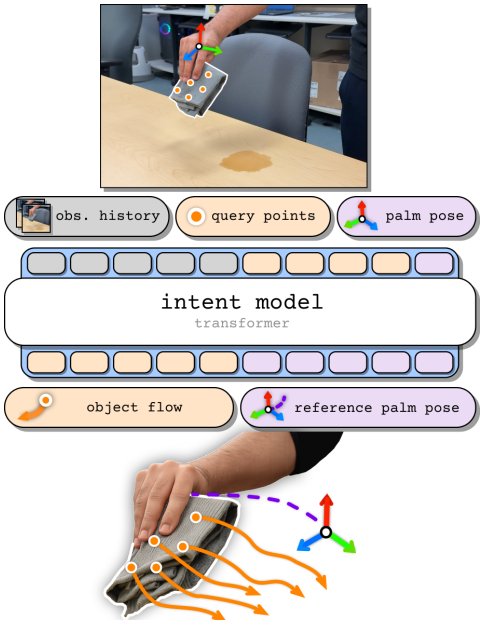

Figure 2: **Intent model.** From the recent observation history, current query points on the object, and the current palm pose, the intent model predicts short-horizon object flow and a reference palm-pose trajectory.

Training. The intent model f_θ is trained with mean-squared-error losses on the flow ($\mathcal{L}_{\text{trk}}$), palm position ($\mathcal{L}_{\text{palm,p}}$), and palm rotation ($\mathcal{L}_{\text{palm,r}}$) streams, averaged over the T future steps and combined as $\mathcal{L}(\theta) = \mathcal{L}_{\text{trk}} + \mathcal{L}_{\text{palm,p}} + \mathcal{L}_{\text{palm,r}}$. Alongside standard photometric and geometric augmentations, we heavily augment the human pixels in $\mathbf{I}_t$ (App. A.2.3) so that f_θ infers object motion without leaning on demonstrator hand appearance, which matters at deploy because the visible hand is the robot, not a human.

Supervision from unstructured video. Per task we mine 20k clips from public video datasets [57, 58, 59, 60, 54] (predominantly in-the-wild YouTube footage), or self-record ∼100 smartphone demonstrations for underrepresented tasks. Each clip is cut into overlapping sliding windows spanning the F past frames (stacked into $\mathbf{I}_t$, ending at $\tau{=}0$) and T future frames ($\tau{=}1,\dots,T$) providing hindsight supervision for f_θ. A four-stage extraction pipeline runs on each window, producing per-frame outputs indexed by τ. ViPE [61] reconstructs per-frame camera intrinsics, extrinsics, and metric depth D_τ. SAM 3.1 [62] produces an object mask M_τ^{obj} and a human mask M_τ^{hum}. DenseTrack3Dv2 [63] samples N query pixels inside M_0^{obj} and tracks them forward; projection through D_τ yields the 3D trajectory $\{\mathbf{x}_{n,\tau}^{\text{trk}}\}$. WiLoR [64] returns a per-frame MANO [65] hand mesh, which we align to D_τ inside M_τ^{hum} via a per-frame rigid fit, yielding the palm pose $(\mathbf{p}_\tau^{\text{palm}}, \mathbf{R}_\tau^{\text{palm}})$. The future-step slice ($\tau{=}1,\dots,T$) of the object flow and palm pose forms the reference $\mathcal{R}$ that supervises f_θ. See App. A.1 for dataset details and the palm-pose lifting procedure.

3.2 Generalist Sensorimotor Policy

The sensorimotor policy π maps onboard sensing to motor commands, and is trained via goal-conditioned RL in Isaac Lab [66] with massively-parallel simulation to realize the intent reference $\mathcal{R}$ on a specific embodiment. We train a separate π per embodiment with the same recipe; this section describes the dexterous-hand setup, with parallel-jaw tweaks in App. C.1.

Setup and training data. We train π to follow references in $\mathcal{R}$'s format, generated procedurally rather than from f_θ or human video. Each episode loads a procedurally generated object (a union of primitives spanning blob-, tool-, and plate-like shapes) under randomized scale, mass, and friction. It then samples a reference trajectory $\mathcal{R}$ of object flow and palm pose: a chain of four segments (approach, in-hand motion, goal, disengage) with randomly sampled grasp, waypoint, and goal poses, exercising grasping, in-hand manipulation, placing, and releasing in one rollout. A per-segment hand-coupling mode varies how tightly the palm tracks the object, so the same object motion admits several valid hand strategies. Training across this broad distribution of objects and

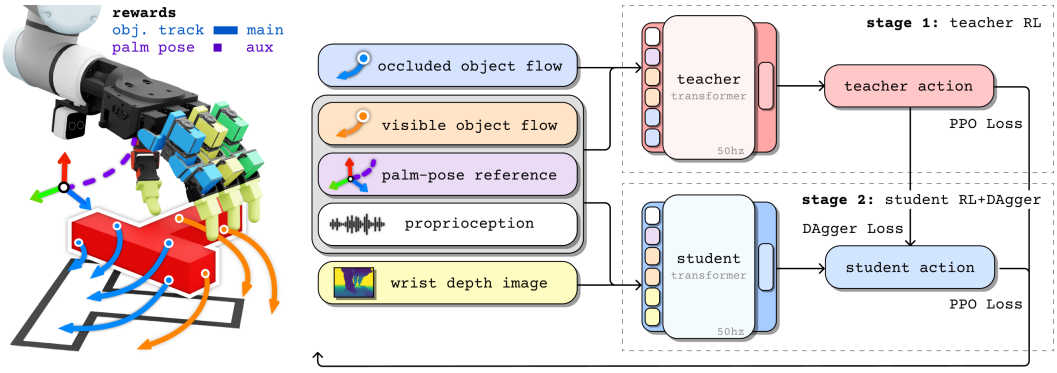

Figure 3: **Sensorimotor policy training.** The teacher π^{T} is first trained with PPO on a privileged sampling of the object-flow component of $\mathcal{R}$ (drawn from the full object surface), the palm-pose reference, and proprioception. The student π^{S} is then distilled from π^{T} with a hybrid PPO + distillation objective, replacing the privileged sampling with the external camera-visible subset of the object flow plus a wrist-mounted depth image.

motions yields a single task-agnostic policy that tracks whatever reference f_θ produces at deploy. Geometry and sampling details are in App. B.1.

Action space. At each step the policy commands the arm and controls the hand through a structured grasp representation. Formally, π outputs $\mathbf{a}_t = [\mathbf{a}_t^{\text{arm}}; \mathbf{a}_t^{\text{eig}}; \mathbf{a}_t^{\text{hnd}}]$: arm joint-position deltas $\mathbf{a}_t^{\text{arm}}$, eigen-grasp coefficients $\mathbf{a}_t^{\text{eig}}$, and per-joint hand residuals $\mathbf{a}_t^{\text{hnd}}$. The basis is fit to retargeted human grasps [67, 23], so $\mathbf{a}_t^{\text{eig}}$ moves the fingers along coordinated modes of natural grasping that bias exploration toward stable grasps, while $\mathbf{a}_t^{\text{hnd}}$ adds per-joint motions the basis cannot express. The action is integrated on top of the previous joint target $\tilde{\mathbf{q}}_{t-1}$, EMA-smoothed, and clipped to joint limits to yield the new target $\tilde{\mathbf{q}}_t$ for the low-level PD controller, following SimToolReal [9]. See App. B.2.

Observations and goal. Both policies receive $\mathcal{R}$ as their goal alongside their state observations. Teacher π^{T} is trained on privileged simulator state and student π^{S} on onboard sensing; both observe the palm-pose component $\{(\mathbf{p}_\tau^{\text{palm}}, \mathbf{R}_\tau^{\text{palm}})\}_{\tau=0}^{T}$ of $\mathcal{R}$ together with short histories of the joint configuration $\mathbf{q}_t$ and the previous joint target $\tilde{\mathbf{q}}_{t-1}$. They differ in how the object-flow component of $\mathcal{R}$ is sampled. π^{T} sees a privileged sampling $\{\bar{\mathbf{x}}_{n,\tau}^{\text{trk}}\}_{n=1,\tau=0}^{\bar{N},T}$ drawn from the full object surface, together with joint velocities $\dot{\mathbf{q}}_t$ and the object pose. π^{S} sees a smaller sampling $\{\mathbf{x}_{n,\tau}^{\text{trk}}\}_{n=1,\tau=0}^{N,T}$ restricted to the surface visible from an external RGB-D camera, together with a wrist-mounted depth image $\mathbf{D}_t^{\text{wrist}}$ that resolves close-range object geometry. The external camera location is randomized per episode during training to cover the viewpoints of the human videos used to train f_θ. All inputs are in the robot base frame; the full table is in App. B.3.

Reward. The sensorimotor policy π is trained with a single task-agnostic reward, balancing three concerns. (1) Object tracking, the primary signal, rewards π for keeping the current query-point positions $\{\bar{\mathbf{x}}_{n,0}^{\text{trk}}\}$ close to the lookahead targets $\{\bar{\mathbf{x}}_{n,\tau}^{\text{trk}}\}_{\tau=1}^{T}$, with a success bonus on reaching the goal pose. (2) Palm-pose following and finger contact enter as shaping rewards, leaving the policy room to deviate where its embodiment requires. (3) Regularization penalizes large action magnitudes $\|\mathbf{a}_t\|$ and unsafe joint configurations. The full inventory, gates, and weights are in App. B.4.

Policy architecture. Both π^{T} and π^{S} build on AME [68], a cross-attention architecture from legged locomotion where a proprioception-conditioned query reads from scene tokens. We adapt it in three ways. (1) The scene tokens come from query points on the object rather than an elevation grid; each query point n becomes one token that encodes its full trajectory $\{\mathbf{x}_{n,\tau}^{\text{trk}}\}_{\tau=0}^{T}$ through a shared point-wise MLP, making the encoder permutation-invariant. (2) π^{S} tokenizes the wrist-depth image $\mathbf{D}_t^{\text{wrist}}$ with a strided-conv stem and joins the resulting patch tokens to the point-trajectory tokens. (3) A self-attention block over the joint token set runs before the cross-attention, letting points and

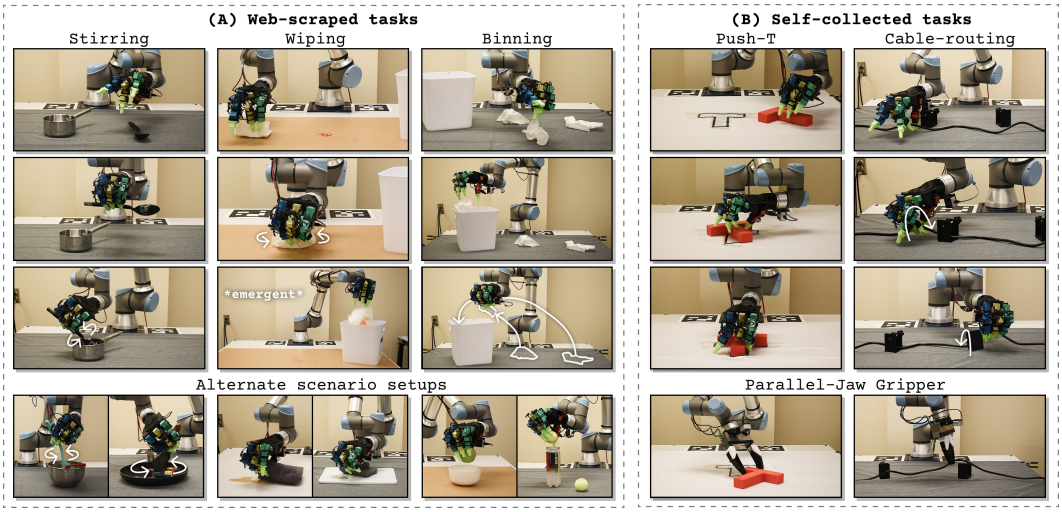

Figure 4: **Real-world tasks** we evaluated: **(A)** Three web-scraped tasks (stirring, wiping, binning), each evaluated under three scenarios. The third wiping panel shows the model depositing the used tissue in a bin without explicit binning supervision. **(B)** Two self-collected tasks (push-T, cable routing), extended to a parallel-jaw gripper setup.

(for π^{S}) depth patches mix locally. The cross-attention output passes through an MLP trunk with proprioception to produce the action distribution and value estimate (Fig. 3, App. B.5).

Training and sim-to-real. The teacher π^{T} is trained with PPO [69] on privileged simulator state under a curriculum that tightens the environment as π^{T} improves. As the curriculum advances, gravity rises from near-zero to full, random object wrench perturbations increase, and success tolerances on object and palm pose tighten. Each environment holds a per-episode difficulty level that rises when π^{T} completes its trajectory within the success tolerance and falls otherwise. The population mean of these levels drives a single scalar $\rho \in [0, 1]$ that interpolates these knobs, following DextrAH-RGB [70, 10] (App. B.6). We also match the simulator's dynamics to the real robot via system identification on the joints (App. B.7).

Following PHP [71], we distill π^{T} into the student policy π^{S} with a hybrid objective $\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{PPO}}(\pi^{\mathrm{S}}) + \lambda_D \left\|\mu^{\mathrm{S}} - \mu^{\mathrm{T}}\right\|_2^2$, combining a PPO surrogate on on-policy rollouts of π^{S} with an MSE regression between the teacher and student action means. The distillation weight λ_D is annealed during training, so imitation dominates early and the on-policy PPO term takes over once π^{S} has closed most of the gap. Throughout distillation the simulator is held at the final curriculum level $\rho = 1$. The student additionally trains under realistic sensor noise on the depth cameras [72] and proprioception, matching the deploy sensors. See App. B.5 for the full loss, student network, and training schedule.

3.3 Real-World Execution

The robot observes through a fixed external RGB-D camera and a wrist-mounted depth camera, running a slow intent cycle around a fast control loop. On each intent cycle (mirroring the training supervision pipeline of §3.1), SAM 3.1 [62] refreshes the object mask, query points are sampled and back-projected to current positions $\{\mathbf{x}_{n,0}^{\text{trk}}\}$, and f_θ is re-queried with these, $\mathbf{I}_t$, and the palm pose $(\mathbf{p}_0^{\text{palm}}, \mathbf{R}_0^{\text{palm}})$ from forward kinematics to produce a fresh $\mathcal{R}$; since f_θ is trained on human video, its outputs arrive in the external-camera frame and are transformed into the robot base frame before use. At every policy step, π^{S} consumes the current points, the flow and palm-pose reference from $\mathcal{R}$, the wrist depth $\mathbf{D}_t^{\text{wrist}}$, and proprioception, and emits $\mathbf{a}_t$. Between intent cycles a sliding-window 3D point tracker [63] keeps the current points consistent with the live scene, while a lookahead window advances over $\mathcal{R}$'s 1 s flow horizon; once it reaches the end, we re-query f_θ. Exact rates are in App. B.8.

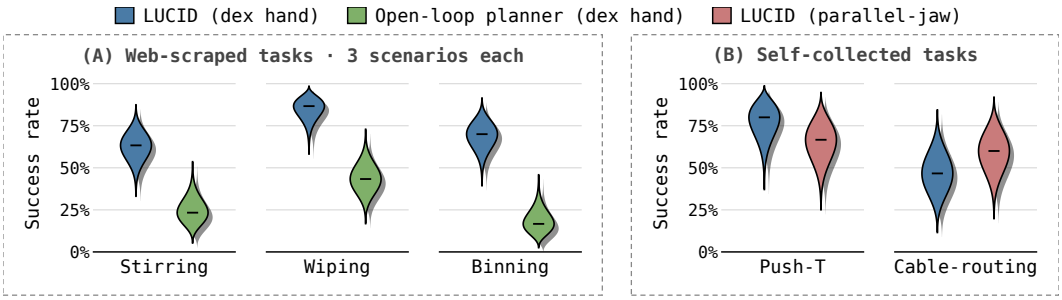

Figure 5: **Real-world success rates.** Per-task success across five real-world tasks, evaluated against task-appropriate baselines. (**A**) LUCID (dex hand) versus an open-loop video-generation planner (dex hand) [73] on web-scale tasks. (**B**) LUCID (dex hand) versus LUCID (parallel-jaw) on self-collected tasks. Failure-mode breakdowns appear in App. C.2.

4 Experimental Results

We investigate four questions about LUCID: (Q1, §4.1) whether the system works end-to-end on web-scraped tasks. (Q2, §4.2) whether scaling human-video supervision improves real-world performance. (Q3, §4.3) whether the intent model transfers across robot embodiments and supports new tasks supervised only by self-collected smartphone video. (Q4, §4.4) which tracking-policy design choices drive performance.

4.1 Real-World Capability on Web-Scraped Tasks

We evaluate three web-scraped tasks, each supervised by 20k human-video clips. (1) Stirring: the robot picks up a spoon and completes three stirring circles inside the container. (2) Wiping: the robot picks up a cloth and clears random marks/substances from a surface. (3) Binning: the robot picks up each object in the workspace and deposits it in a target container. Each task is evaluated over 10 trials in each of three scenarios that jointly vary object instances, table setup, and external camera pose, all out-of-distribution for the intent model. To isolate the contribution of closed-loop intent, we compare against an open-loop planner: Veo 3.1 [73] generates a single video from the initial RGB observation, whose extracted object flow and palm-pose reference drive the same sensorimotor policy (App. C.1).

LUCID's strength is closed-loop intent: when the initial grasp misses or the object shifts mid-rollout, the intent model re-queries the scene and redirects the policy (Fig. 5**A**; per-scenario in App. C.2). The sensorimotor policy also handles deformable objects (tissue, towel, and cloth) despite being trained only on rigid objects in simulation. We also observe emergent task composition: when we place a bin near the workspace after the robot wipes ketchup with a tissue, the intent model picks up the soiled tissue and deposits it in the bin (Fig. 4A, Wiping), without any task-specific binning supervision; we attribute this to incidental binning in the broader 20k-clip wiping pool. Failures cluster into perception issues from object occlusion (e.g., the spoon being covered by the hand) and unrecoverable states from unstable grasps (e.g., tennis balls slipping off the table). The open-loop baseline fails differently when execution deviates from the generated plan. For example, a misgrasp during stirring sends the spoon flipping to a new location, and the stale references confuse the sensorimotor policy (App. C.1). Veo 3.1 can also hallucinate scene details, e.g., wiped regions where no wipe occurred.

4.2 Intent Data Scaling

We sweep the binning training corpus across $\{1\text{k}, 2\text{k}, 5\text{k}, 10\text{k}, 20\text{k}\}$ clips. At each scale point we run 10 real-world trials in each of the three binning scenarios from §4.1 (30 trials per scale point) and evaluate intent loss on a held-out set of 1k binning clips. Held-out intent loss decreases steadily with corpus size (well-fit by a power law over the observed range, see App. C.3), and real-world success on binning rises with it (Fig. 6). At 1k–2k clips the intent model produces poor references (e.g., the

policy reaches the object but fails to identify the bin); in the 5k–10k range, container localization emerges but placement alignment remains weak, often missing the bin on release.

4.3 Intent Transfer Across Robot Embodiments

To test intent transfer across embodiments, we evaluate (1) push-T [74]: the robot pushes a T-shaped block to a target pose, and (2) cable routing: the robot threads a cable through two fixtures (Fig. 4**B**). For each task, the intent model is trained on 1 hr of self-collected smartphone video. We deploy the same intent predictions with two sensorimotor policies trained in simulation: our primary LEAP hand policy, and a parallel-jaw gripper variant with minor embodiment-specific tweaks (App. C.1). We run 15 real-world trials per embodiment per task; per-scenario breakdowns in App. C.2.

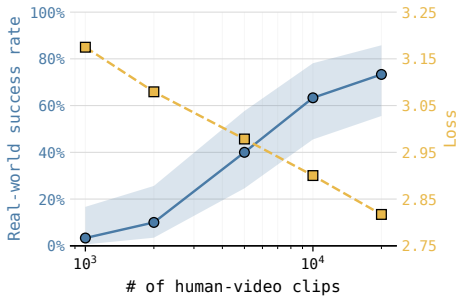

Figure 6: **Intent data scaling.** Sweeping intent-model training data from 1k to 20k human-video clips on the binning task, real-world success rises and held-out intent loss falls.

The hand and gripper policies reach the same aggregate success on these two tasks despite very different morphologies (19/30 each; Fig. 5**B**). Cable routing actually favors the gripper, since two opposing jaws are well-suited to grasping the thin cable, while the dexterous hand has a difficult time precisely grasping it. Failures on both embodiments concentrate on out-of-distribution states where the intent model's predictions fail to drive the policy forward, likely due to the small 1 hr smartphone corpus.

4.4 Sensorimotor Policy Ablations

Four ablations probe the choices that most shaped `LUCID`'s recipe, run in simulation with 3 seeds each. On the teacher (Fig. 7A) we (1) replace our policy encoder with an MLP encoder concatenating all inputs without tokenization, and (2) drop the eigen-grasp basis, replacing it with per-joint actions only. On the student (Fig. 7B) we (3) replace our hybrid distillation objective [71] with a DAgger-BC mixture [75] over a 50/50 teacher/student rollout blend, and (4) remove the wrist camera.

The MLP encoder still localizes the object but loses the per-point detail required for stable grasp contacts and in-hand manipulation. Without the eigen-grasp basis the policy faces a much larger exploration space; with the basis, training first exploits the eigen modes for natural-looking grasps and later refines individual joints for finer manipulation. DAgger-BC is a competitive baseline, but because it only mimics the teacher the student cannot fully exploit its own input modalities (e.g., the wrist camera). Removing the wrist camera leaves object geometry unresolved, similar to the MLP encoder ablation. See App. C.4 for an additional ablation on query-point count N.

5 Limitations

(1) Pipeline brittleness. The system is over-modularized; SAM 3.1, DenseTrack3Dv2, ViPE, and WiLoR each introduce a failure point during data extraction and at deployment. Perception faults like in-hand occlusion or textureless objects propagate down the chain to the sensorimotor policy. Training the intent model end-to-end from raw video would replace this chain with a single learned representation. **(2) Task-condition gap.** Tasks without a verifiable end condition cause the intent model to loop indefinitely (stirring currently requires an externally imposed stop), and training on a new task requires manual corpus filtering, which scales poorly. Scaling to thousands of tasks through text conditioning would address both, since a high-level planner could prompt the intent model ("stir", then "put spoon down") to stitch sub-tasks. **(3) Lossy explicit interface.** The 3D-flow-plus-palm-pose interface between f_θ and π^{S} is hand-designed and discards information such as finger

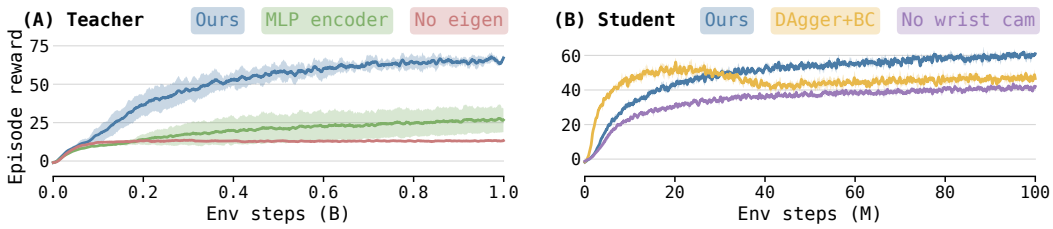

Figure 7: **Sensorimotor policy ablations.** Episode reward against environment steps for the teacher training **(A)** and the student distillation **(B)**. (**A**): Ours versus an MLP encoder concatenating all inputs and per-joint actions without the eigen-grasp basis. (**B**): Ours versus DAgger-BC distillation and no wrist camera.

configuration and fine contact. Even when intent is predicted accurately, it omits cues the policy would need to fully reproduce the human demonstration. A latent intermediate jointly optimized across both sub-models would let the interface adapt to whatever signal the policy needs [76].

6 Conclusion

We presented LUCID, a framework that separates robot-agnostic intent (supervised by unstructured human video) from embodiment-specific control (learned in massively-parallel simulation). It succeeds on real-world tasks from both internet and smartphone video, scales with supervision volume, and transfers to a parallel-jaw gripper. Pairing two independently scalable supervision sources is, in our view, a promising direction for dexterous manipulation that scales without teleoperation.

Acknowledgments

Research supported by the NVIDIA Academic Grant Program using NVIDIA Brev. Guanya Shi holds concurrent appointments as an Assistant Professor at Carnegie Mellon University and as an Amazon Scholar. This paper describes work performed at Carnegie Mellon University and is not associated with Amazon.

References

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023. doi:10.15607/RSS.2023.XIX.016.

[2] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi:10.1109/ICRA57147.2024.10611477.

[3] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[4] I. Guzey, H. Qi, J. Urain, C. Wang, J. Yin, K. Bodduluri, M. Lambeta, L. Pinto, A. Rai, J. Malik, T. Wu, A. Sharma, and H. Bharadhwaj. Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations. *arXiv preprint arXiv:2511.16661*, 2025.

[5] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi:10.15607/RSS.2024.XX.045.

[6] H. Gupta, X. Guo, H. Ha, C. Pan, M. Cao, D. Lee, S. Scherer, S. Song, and G. Shi. UMI-on-Air: Embodiment-aware guidance for embodiment-agnostic visuomotor policies. *arXiv preprint arXiv:2510.02614*, 2025.

[7] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2475–2499. PMLR, 2025. URL `https://proceedings.mlr.press/v270/xu25a.html`.

[8] H. Li, L. Sun, Y. Hu, D. Ta, J. Barry, G. Konidaris, and J. Fu. NovaFlow: Zero-shot manipulation via actionable flow from generated videos. *arXiv preprint arXiv:2510.08568*, 2025.

[9] K. Kedia, T. G. W. Lum, J. Bohg, and C. K. Liu. SimToolReal: An object-centric policy for zero-shot dexterous tool manipulation. *arXiv preprint arXiv:2602.16863*, 2026.

[10] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. Van Wyk. DextrAH-RGB: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024.

[11] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. DexMV: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision (ECCV)*, 2022.

[12] H. Gupta, M. A. Mirzaee, and W. Yuan. Grasp to act: Dexterous grasping for tool use in dynamic settings. *arXiv preprint arXiv:2602.20466*, 2026.

[13] Y. Kuang, S. Park, K. Fragkiadaki, and S. Tulsiani. Dex4D: Task-agnostic point track policy for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2602.15828*, 2026.

[14] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, Q. Ma, S. Nah, L. Magne, J. Xiang, Y. Xie, R. Zheng, D. Niu, Y. L. Tan, K. R. Zentner, G. Kurian, S. Indupuru, P. Jannaty, J. Gu, J. Zhang, J. Malik, P. Abbeel, M.-Y. Liu, Y. Zhu, J. Jang, and L. Fan. DreamDojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

[15] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, A. Malik, K. Lee, W. Liang, N. Ranawaka, J. Gu, Y. Xu, G. Wang, F. Hu, A. Narayan, J. Bjorck, J. Wang, G. Kim, D. Niu, R. Zheng, Y. Xie, J. Wu, Q. Wang, R. Julian, D. Xu, Y. Du, Y. Chebotar, S. Reed, J. Kautz, Y. Zhu, L. Fan, and J. Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[16] X. Liu, J. Adalibieke, Q. Han, Y. Qin, and L. Yi. DexTrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *International Conference on Learning Representations (ICLR)*, 2025.

[17] S. Xu, Y.-W. Chao, L. Bian, A. Mousavian, Y.-X. Wang, L. Gui, and W. Yang. Dexplore: Scalable neural control for dexterous manipulation from reference scoped exploration. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 2184–2199. PMLR, 2025. URL https://proceedings.mlr.press/v305/xu25d.html.

[18] K. Shaw, A. Agarwal, and D. Pathak. LEAP Hand: Low-cost, efficient, and anthropomorphic hand for robot learning. In *Robotics: Science and Systems (RSS)*, 2023.

[19] Z. Chen, S. Chen, E. Arlaud, I. Laptev, and C. Schmid. ViViDex: Learning vision-based dexterous manipulation from human videos. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[20] J. Hsieh, K.-H. Tu, K.-H. Hung, and T.-W. Ke. DexMan: Learning bimanual dexterous manipulation from human and generated videos. *arXiv preprint arXiv:2510.08475*, 2025.

[21] J. Mu, S. Yang, Y. Bao, H. Bae, T. Wei, L. Xu, B. Li, H. Xu, and J. Pang. DexImit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

[22] H. Chen, T. Dong, T. Wu, L. Wang, Y. Jangir, Y. Niu, Y. Ye, H. Bharadhwaj, Z. Erickson, and J. Ichnowski. Dexterous manipulation policies from RGB human videos via 3D hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026.

[23] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg. Crossing the human-robot embodiment gap with sim-to-real RL using one human demonstration. *arXiv preprint arXiv:2504.12609*, 2025.

[24] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan. SPIDER: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.

[25] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. ZeroMimic: Distilling robotic manipulation skills from web videos. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 16939–16947, 2025. doi:10.1109/ICRA55743.2025.11128283.

[26] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 4545–4565. PMLR, 2025. URL https://proceedings.mlr.press/v305/lepert25a.html.

[27] Z. Wang, B. He, K. Yu, S. Lee, R. Gao, F. Huang, and Y. Aloimonos. HumanEgo: Zero-shot robot learning from minutes of human egocentric videos. *arXiv preprint arXiv:2605.24934*, 2026.

[28] S. Haldar and L. Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

[29] I. Guzey, Y. Dai, G. Savva, R. Bhirangi, and L. Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3344–3351, 2025. doi:10.1109/ICRA55743.2025.11128690.

[30] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *Computer Vision – ECCV 2024*, volume 15134 of *Lecture Notes in Computer Science*, pages 306–324, 2024. doi:10.1007/978-3-031-73116-7_18.

[31] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3943–3960. PMLR, 2025. URL https://proceedings.mlr.press/v270/liang25b.html.

[32] J. Pai, L. Achenbach, V. Montesinos, B. Forrai, O. Mees, and E. Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

[33] R. G. Goswami, A. Bar, D. Fan, T.-Y. Yang, G. Zhou, P. Krishnamurthy, M. Rabbat, F. Khorrami, and Y. LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025.

[34] S. Routray, H. Pan, U. Jain, S. Bahl, and D. Pathak. ViPRA: Video prediction for robot actions. In *International Conference on Learning Representations (ICLR)*, 2026.

[35] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[36] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, G. Yang, J. Zhang, S. Yi, G. Shi, and X. Wang. Humanoid policy $\sim$ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[37] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, X. Cheng, R.-Z. Qiu, H. Yin, S. Liu, S. Han, Y. Lu, and X. Wang. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[38] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. EgoMimic: Scaling imitation learning via egocentric video. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[39] Q. Li, Y. Deng, Y. Liang, L. Luo, L. Zhou, C. Yao, L. Zeng, Z. Feng, H. Liang, S. Xu, Y. Zhang, X. Chen, H. Chen, L. Sun, D. Chen, J. Yang, and B. Guo. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

[40] M. Lepert, J. Fang, and J. Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025.

[41] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8802–8810, 2025. doi: 10.1109/ICRA55743.2025.11128834.

[42] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg. AMPLIFY: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025.

[43] X. Liu, K. Lyu, J. Zhang, T. Du, and L. Yi. Parameterized quasi-physical simulators for dexterous manipulations transfer. In *Computer Vision – ECCV 2024*, volume 15136 of *Lecture Notes in Computer Science*, pages 164–182, 2024. doi:10.1007/978-3-031-73229-4_10.

[44] S. Dasari, A. Gupta, and V. Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[45] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[46] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song. DexMachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

[47] Z.-H. Yin, C. Wang, L. Pineda, F. Hogan, K. Bodduluri, A. Sharma, P. Lancaster, I. Prasad, M. Kalakrishnan, J. Malik, M. Lambeta, T. Wu, P. Abbeel, and M. Mukadam. DexterityGen: Foundation controller for unprecedented dexterity. *arXiv preprint arXiv:2502.04307*, 2025.

[48] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024.

[49] C. Yuan, C. Wen, T. Zhang, and Y. Gao. General flow as foundation affordance for scalable robot learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 1541–1566. PMLR, 2025. URL https://proceedings.mlr.press/v270/yuan25a.html.

[50] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held. ToolFlowNet: Robotic manipulation with tools via predicting tool flow from point clouds. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1038–1049. PMLR, 2023. URL https://proceedings.mlr.press/v205/seita23a.html.

[51] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations (ICLR)*, 2025.

[52] W. Huang, Y.-W. Chao, A. Mousavian, M.-Y. Liu, D. Fox, K. Mo, and L. Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

[53] P. Mandikal and K. Grauman. DexVIP: Learning dexterous grasping with human hand pose priors from video. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 651–661. PMLR, 2022. URL https://proceedings.mlr.press/v164/mandikal22a.html.

[54] B. Chen, T. Zhang, H. Geng, K. Song, C. Zhang, P. Li, W. T. Freeman, J. Malik, P. Abbeel, R. Tedrake, V. Sitzmann, and Y. Du. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025.

[55] N. Karaev, Y. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6013–6022, 2025.

[56] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[57] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, and S. Tulyakov. Panda-70M: Captioning 70M videos with multiple cross-modality teachers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[58] D. Chen, T. Kasarla, Y. Bang, M. Shukor, W. Chung, J. Yu, A. Bolourchi, T. Moutakanni, and P. Fung. Action100M: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026.

[59] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[60] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. doi:10.1007/s11263-021-01531-2.

[61] J. Huang, Q. Zhou, H. Rabeti, A. Korovko, H. Ling, X. Ren, T. Shen, J. Gao, D. Slepichev, C.-H. Lin, J. Ren, K. Xie, J. Biswas, L. Leal-Taixé, and S. Fidler. ViPE: Video pose engine for 3D geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

[62] 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. *arXiv preprint arXiv:2511.16719*, 2025.

[63] T. D. Ngo, A. Mirzaei, G. Qian, H. Liang, C. Gan, E. Kalogerakis, P. Wonka, and C. Wang. DELTAv2: Accelerating dense 3D tracking. *arXiv preprint arXiv:2508.01170*, 2025.

[64] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[65] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17, 2017. doi:10.1145/3130800.3130883.

[66] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrügg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makoviychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg, C. Schwarke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. Van Wyk, F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, C. Chen, N. Jawale, A. V. Kuruttukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche,

N. Moënne-Loccoz, S. Park, R. Stepinski, D. Van Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. d. H. Ciechomski, G. Daviet, M. Mohajerani, J. von Muralt, V. Reutskyy, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State. Isaac lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[67] M. Ciocarlie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3270–3275, 2007. doi:10.1109/IROS.2007.4399227.

[68] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter. Attention-based map encoding for learning generalized legged locomotion. *Science Robotics*, 10(105):eadv3604, 2025. doi:10.1126/scirobotics.adv3604.

[69] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[70] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[71] Z. Wu, X. Huang, L. Yang, Y. Zhang, K. Sreenath, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferrazza, G. Shi, and C. K. Liu. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching. *arXiv preprint arXiv:2602.15827*, 2026.

[72] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014. doi:10.1109/ICRA.2014.6907054.

[73] Google DeepMind. Veo 3 model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf, 2026. Published May 23, 2025; updated January 13, 2026. Accessed: 2026-06-05.

[74] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. M. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi:10.15607/RSS.2023.XIX.026.

[75] T. He, Z. Wang, H. Xue, Q. Ben, Z. Luo, W. Xiao, Y. Yuan, X. Da, F. Castañeda, S. Sastry, C. Liu, G. Shi, L. Fan, and Y. Zhu. VIRAL: Visual sim-to-real at scale for humanoid loco-manipulation. *arXiv preprint arXiv:2511.15200*, 2025.

[76] R. S. Sutton. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, 2019.

[77] D. Makoviichuk and V. Makoviychuk. RL Games: High performance RL library. https://github.com/Denys88/rl_games, 2021.

[78] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi:10.1162/106365601750190398.

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

A Intent Model Details

A.1 Supervision Pipeline

This appendix details how each raw video clip is processed into the per-window supervision targets consumed by the intent-model loss (Sec. 3.1).

A.1.1 Dataset Mix and Clip Extraction

Per task we mine 20k clips from a mix of Panda-70M [57], Action100M [58], Something-Something-V2 [59], EPIC-Kitchens [60], and the LVP [54] metadata release (filtered by action label and length; predominantly in-the-wild YouTube footage), or self-record on a mounted iPhone 16 Pro Max when the task is underrepresented in those datasets. The only annotation is the object name used to prompt segmentation. Clips are resampled to 8 Hz and cut into sliding windows at stride 2, following the window structure of Sec. 3.1.

A.1.2 Extraction Pipeline

Each window is processed by four stages, illustrated in Fig. 8.

a) **Camera and depth reconstruction.** ViPE [61] runs monocular SLAM and metric depth estimation, returning per-frame intrinsics $\mathbf{K}_\tau$, extrinsics $\mathbf{E}_\tau \in SE(3)$, and a dense depth map D_τ.

b) **Object and hand segmentation.** SAM 3.1 [62], prompted with the action noun, produces a per-frame object mask M_τ^{obj} and a separate human mask M_τ^{hum}. We subtract the human mask from the object mask, $M_\tau^{\text{obj}} \leftarrow M_\tau^{\text{obj}} \setminus M_\tau^{\text{hum}}$, so hand pixels never leak into the object region.

c) **Object flow tracks.** DenseTrack3Dv2 [63] samples N query pixels inside M_0^{obj} and tracks them across the window. We back-project each tracked pixel through D_τ and $\mathbf{K}_\tau$ to obtain the 3D trajectory $\{\mathbf{x}_{n,\tau}^{\text{trk}}\}$.

d) **Hand reconstruction.** WiLoR [64] returns a MANO [65] hand mesh with wrist rotation $\mathbf{R}_\tau^{\text{W}} \in SO(3)$, in a hand-local frame at arbitrary scale. A per-frame rigid fit (Sec. A.1.3) aligns the MANO mesh to D_τ inside M_τ^{hum} and reads off the palm pose $(\mathbf{p}_\tau^{\text{palm}}, \mathbf{R}_\tau^{\text{palm}}) \in SE(3)$ at the MANO palm-center vertex.

The intent-model supervision targets per window are the 3D query-point trajectory $\{\mathbf{x}_{n,\tau}^{\text{trk}}\}$ and the palm-pose trajectory $\{(\mathbf{p}_\tau^{\text{palm}}, \mathbf{R}_\tau^{\text{palm}})\}$ over the T future steps, which together form the reference $\mathcal{R}$ of Sec. 3.1.

A.1.3 Palm-Pose Lifting

For each frame τ we solve a uniform scale $s \in \mathbb{R}_{>0}$ and translation $\mathbf{t} \in \mathbb{R}^3$ that align the MANO mesh with the ViPE depth at the hand pixels:

$$\min_{s,\mathbf{t}} \sum_{i \in \mathcal{V}_\tau} \left\| s\, \mathbf{v}_{\tau,i}^{\text{M}} + \mathbf{t} \;-\; \text{unproj}\big(\mathbf{u}_i,\, \bar{d}_\tau;\, \mathbf{K}_\tau\big) \right\|_2^2, \tag{1}$$

where $\mathcal{V}_\tau$ is the set of back-face-culled MANO vertices filtered by M_τ^{hum}, each projected to pixel $\mathbf{u}_i$, and $\bar{d}_\tau$ is the median ViPE depth over those pixels after outlier rejection. The objective is linear in (s, t_x, t_y) and closed-form in t_z, and each frame reduces to a single `lstsq` call. We read off the palm pose as $\mathbf{p}_\tau^{\text{palm}} = s\, \mathbf{v}_\tau^{\text{palm}} + \mathbf{t}$ using the MANO palm-center vertex, and $\mathbf{R}_\tau^{\text{palm}} = \mathbf{R}_\tau^{\text{W}}\, \mathbf{R}_{\text{WM}}$ with a fixed hand-to-palm change-of-basis $\mathbf{R}_{\text{WM}}$.

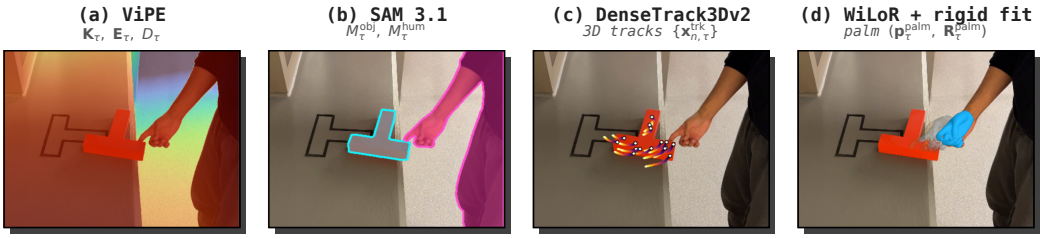

Figure 8: **Supervision extraction pipeline.** Each video window is processed by four stages: **(a)** ViPE [61] for camera intrinsics, extrinsics, and metric depth; **(b)** SAM 3.1 [62] for object and human masks; **(c)** DenseTrack3Dv2 [63] for 3D object-flow tracks; and **(d)** WiLoR [64] with a rigid fit (Eq. 1) for the palm pose. See App. A.1.2 for full details.

A.2 Architecture and Training Configuration

A.2.1 Architecture

The intent model f_θ ingests a stack of F past RGB-D frames and produces T-step predictions for N object tokens and a single palm token. Core sizing is in Table 1.

Encoder. A frozen ViT-B/16 DINOv3 [56] encodes RGB into patch tokens. A zero-initialized residual adapter (Conv2d $\rightarrow$ LayerNorm $\rightarrow$ Linear) fuses depth into the same token space.

Transformer. The patch tokens enter an EfficientUpdateFormer from CoTracker3 [55], running forward-in-time. Each block applies cross-attention from query tokens into the scene patch tokens, followed by spatial self-attention across the $N{+}1$ query tokens at a given future step and temporal self-attention across the T future steps for each token. Each object token is initialized from its query point $\mathbf{x}_{n,0}^{\text{trk}}$ and the palm token from $(\mathbf{p}_0^{\text{palm}}, \mathbf{R}_0^{\text{palm}})$, both encoded sinusoidally and projected to the transformer hidden size.

Outputs. Linear heads at each future step produce object-point displacements, palm-position displacements, and relative palm-rotation displacements. These are composed with the current query points and current palm pose to recover the future object positions and palm poses. Training minimizes mean-squared error between these outputs and the supervision trajectories of App. A.1.

A.2.2 Training

We train a separate intent model per task. Targets are per-channel standardized using statistics from the training set, and weights are tracked by an exponential moving average used for evaluation. All training hyperparameters are in Table 2. Both the supervision extraction (App. A.1.2) and intent-model training run on a A100 GPU node.

A.2.3 Augmentations

Standard geometric and photometric augmentations (random rotation, translation, flips, color jitter, and Gaussian noise on RGB and depth) are applied to $\mathbf{I}_t$ with probability $p_{\text{aug}} = 0.8$. Human-region augmentations are applied with probability $p_{\text{human}} = 0.6$ and replace the pixels inside M_τ^{hum} with one of four appearance modes drawn by a fixed mixture: heavy color jitter (0.40), uniform random RGB fill (0.25), a 2–6 cell color-patch grid (0.25), or per-pixel random RGB (0.10); a brightness gradient and depth-pixel noise are applied on top.

Table 1: Intent model architecture.

Hyperparameter	Value
Inputs	
Input resolution	256×256
Frame stack F	2
Object query tokens N	16
Future steps T	8
Prediction horizon	1 s
Backbone	
RGB encoder	ViT-B/16 DINOv3 (frozen)
Patch size	16×16
Hidden size	768
Tracking transformer	
Backbone	EfficientUpdateFormer
Depth	12
Hidden size	768
Attention heads	12
Factorization	spatial / temporal / scene cross-attention
Output heads	
Object head	Linear($768 \rightarrow 3$) per future step
Palm head	Linear($768 \rightarrow 9$) per future step

Table 2: Intent model training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
Epochs	100
Batch size	16
Learning rate	1×10^{-4}
Minimum learning rate	1×10^{-6}
LR schedule	cosine
Warmup epochs	5
Weight decay	0.01
Gradient clip (norm)	1.0
EMA decay	0.999

B Sensorimotor Policy Details

B.1 Procedural objects and trajectories

At environment reset we sample an object instance and a reference trajectory that specifies how the object and the palm should move through the episode.

Shape pool. The object pool (Fig. 9) is generated offline by sampling small sets of primitives (boxes, spheres, capsules, cylinders, and flat plates), attaching them at random surface points, and taking the boolean union. This covers compact blob-like, elongated tool-like, and flat book-like silhouettes. Each environment uniformly samples one asset and a scale multiplier in $[0.65, 1.1]$; mass, friction, and other properties follow App. B.6.

Reference trajectory. A trajectory is a chain of four *segments*, each defined by an object target pose, a palm target pose, a duration, and a hand-coupling mode. Within a segment, the object and palm ease smoothly from the previous segment's endpoint to the current segment's target over the assigned duration. The hand-coupling mode sets how tightly the palm tracks the object during the segment: *full* rigidly locks palm position and rotation to the object, *position-only* releases rotation, and *none* decouples the palm entirely. The sensorimotor policy sees the next T reference steps at the 8 Hz reference-step rate over the 1 s horizon emitted by f_θ at deploy. Table 3 lists the four segments.

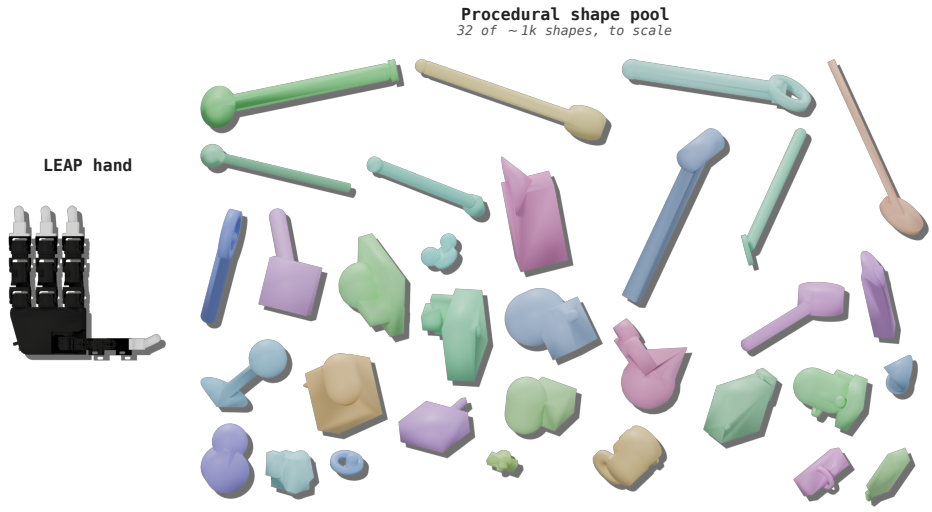

Figure 9: **Procedural shape pool.** 32 random samples from the ∼1k-shape pool used to train π, drawn to scale beside the LEAP hand. Generation details in App. B.1.

Table 3: Reference-trajectory segment schema. Durations in seconds.

Segment	Object motion	Duration (s)	Hand coupling
Approach	static at reset pose	2.5	full
In-hand motion	0–2 random waypoints	0.0–7.5	full or position-only (0.5/0.5)
Goal	single waypoint	3.0	full or position-only (0.5/0.5)
Disengage	static at goal	2.5	none

Segment endpoint sampling. The *approach* segment ends at a grasp location sampled on the object surface, offset along the outward normal by a short standoff; contact points on the bottom half of the surface, approaches pointing toward the robot base, and upward-facing orientations are rejected. The *in-hand* segment inserts 0–2 random object waypoints in a bounded workspace with random per-waypoint rotations, and at each waypoint the palm-in-object frame is optionally perturbed so that the same object trajectory admits several valid hand strategies. The *goal* segment ends at an object pose sampled uniformly in the workspace with random yaw, and the *disengage* segment retreats the palm away from the goal.

B.2 Action space

The action channels $\mathbf{a}_t^{\text{arm}} \in \mathbb{R}^6$, $\mathbf{a}_t^{\text{hnd}} \in \mathbb{R}^{16}$, and $\mathbf{a}_t^{\text{eig}} \in \mathbb{R}^5$ are tanh-squashed continuous outputs.

Eigen-grasp basis. The basis $\mathbf{E} \in \mathbb{R}^{16\times 5}$ is the top-5 principal-component basis of a dataset of retargeted human grasps [23], and the combined hand delta is $\Delta\mathbf{q}_t^{\text{hand}} = \mathbf{E}\mathbf{a}_t^{\text{eig}} + \mathbf{a}_t^{\text{hnd}}$.

Integration. Following SimToolReal [9], the raw target is $\mathbf{q}_t^{\text{raw}} = \tilde{\mathbf{q}}_{t-1} + \eta\,[\mathbf{a}_t^{\text{arm}};\, \Delta\mathbf{q}_t^{\text{hand}}]$ and the PD setpoint is the EMA $\tilde{\mathbf{q}}_t = \boldsymbol{\alpha} \odot \mathbf{q}_t^{\text{raw}} + (\mathbf{1} - \boldsymbol{\alpha}) \odot \tilde{\mathbf{q}}_{t-1}$ clipped to joint limits, where $\boldsymbol{\alpha}$ is a per-joint vector with $\alpha^{\text{arm}} = 0.15$ on the arm block and $\alpha^{\text{hnd}} = 0.75$ on the hand block, and $\eta = 0.1$. At reset $\tilde{\mathbf{q}}$ is initialized to the current joint configuration so the policy begins as a residual on current state.

B.3 Observation spaces

Table 4 lists the observations consumed by π^{T} and π^{S}. $\mathbf{c}_t \in \mathbb{R}^5$ is the current hand-joint offset from the default pose projected onto $\mathbf{E}$ (App. B.2). The segment progress scalar is an internal clock that

Table 4: Observations consumed by the teacher and student policies.

Observation	π^{T}	π^{S}	**Dim**	**History**
Previous action $\mathbf{a}_{t-1}$	✓	✓	27	8
Joint positions $\mathbf{q}_t$	✓	✓	22	8
Joint velocities $\dot{\mathbf{q}}_t$	✓		22	8
Previous joint target $\tilde{\mathbf{q}}_{t-1}$	✓	✓	22	8
Eigen-grasp projection $\mathbf{c}_t$	✓	✓	5	8
Palm and fingertip body states	✓		65	8
Fingertip contact forces	✓		12	8
Mid-phalanx contact forces	✓		9	8
Object pose (palm frame)	✓		7	8
Object pose (world)	✓		7	8
Object-pose lookahead	✓		$7T$	1
Object-pose tracking error	✓		7	1
Palm pose $(\mathbf{p}_0^{\text{palm}}, \mathbf{R}_0^{\text{palm}})$	✓	✓	7	8
Palm-pose lookahead $(\mathbf{p}_\tau^{\text{palm}}, \mathbf{R}_\tau^{\text{palm}})$	✓	✓	$7T$	1
Palm-pose tracking error	✓		7	1
Full-surface query points $\bar{\mathbf{x}}_{n,0}^{\text{trk}}$	✓		$3\bar{N}$	4
Full-surface lookahead $\bar{\mathbf{x}}_{n,\tau}^{\text{trk}}$	✓		$3\bar{N}T$	1
Camera-visible query points $\mathbf{x}_{n,0}^{\text{trk}}$		✓	$3N$	1
Camera-visible lookahead $\mathbf{x}_{n,\tau}^{\text{trk}}$		✓	$3NT$	1
Segment progress scalar	✓	✓	1	1
Wrist depth image $\mathbf{D}_t^{\text{wrist}}$		✓	80×60	1

ramps from 0 to 1 between consecutive reference targets, telling the policy where it is within the current interpolation.

B.4 Reward composition

The sensorimotor policy is trained with a single task-agnostic reward (Table 6). Each episode is split into three phases aligned with the trajectory segments of App. B.1: $G =$ approach, $M =$ in-hand motion + goal, $R =$ disengage. Throughout, (e_p, e_θ) denote object position/rotation tracking errors and (e_p^h, e_θ^h) the palm counterparts; position values are in meters and rotation values in radians, including inside the exp and tanh kernels.

Two shared gates appear in many rows. The *contact factor* $c(\mathbf{f})$ gates rewards that should only pay out during a thumb-opposed grasp, and adds a small capped bonus when the palm and proximal finger links also contact the object. Using the saturation kernel $s(\cdot)$ of Table 5, with $s_i = s(\|\mathbf{f}_i\|)$ per fingertip and $s_{\text{palm}} = s(F_{\text{palm}})$,

$$\begin{aligned} c(\mathbf{f}) &= s_{\text{thumb}} \cdot \big(2.0\, s_{(0)} + 1.8\, s_{(1)} + 1.4\, s_{(2)}\big) \;+\; 0.4\, s_{\text{palm}}, \\ &s_{(k)} \text{ is the } k\text{-th largest of } \{s_{\text{index}}, s_{\text{middle}}, s_{\text{ring}}\}. \end{aligned} \tag{2}$$

The *palm-pose gate* g_{hp} gates tracking rewards on palm alignment,

$$g_{\text{hp}} = \min\big(r(e_p^h;\, \delta_p^{\text{hp}}),\; r(e_\theta^h;\, \delta_\theta^{\text{hp}})\big), \qquad r(x;[a,b]) = \text{clip}((b-x)/(b-a), 0, 1), \tag{3}$$

with ramps $\delta_p^{\text{hp}} = (0.04, 0.06)$ and $\delta_\theta^{\text{hp}} = (0.4, 0.6)$.

The remaining symbols used in Table 6 are collected in Table 5.

Termination. An episode terminates early and pays the termination penalty if the object or palm pose deviates from its reference beyond an ADR-interpolated threshold (App. B.6) or if any joint enters a non-physical configuration. A natural termination occurs at the end of the disengage segment and pays the timeout bonus.

Table 5: Symbols used in Table 6.

Symbol	Meaning
Phase	
$\phi \in \{G, M, R\}$	current phase.
Fingertip contact	
$s(x)$	$\mathrm{clip}((x-0.5)/1.5, 0, 1)$ kernel applied to per-body force magnitudes.
$\mathbf{f}_i$	force vector at fingertip $i \in \{\text{thumb, index, middle, ring}\}$.
F_{palm}	summed force over palm and proximal-finger links.
$F_{\text{tot}} = \sum_i \lVert\mathbf{f}_i\rVert$	total fingertip force.
$g_{\text{cont}}(F_{\text{tot}})$	contact-required gate, $0.05 \to 1$ as F_{tot} ramps $1 \to 2$ N.
$\mathbf{f}_b^{\text{net}}, \mathbf{f}_b^{\text{obj}}$	net and object-filter contact forces on self-contact link $b \in \mathcal{B}_{\text{sc}}$.
Object tracking	
e_t	scalar aggregate of object tracking error at time t.
$e_{t-\Delta}$	same error one reference step earlier.
Δ	one reference step (8 Hz over the 1 s horizon emitted by f_θ).
$\Delta\mathbf{p}$	object-in-palm position change over Δ.
$\Delta\theta$	object-in-palm rotation change over Δ.
$(\sigma_p, \sigma_\theta) = (0.03, 0.4)$	palm-pose-following kernel widths (m, rad).
(τ_p, τ_θ)	trajectory-success thresholds, ADR $(0.07, 0.7) \to (0.04, 0.4)$.
$\mathbf{v}_{\text{obj}}$	object linear velocity.
$\boldsymbol{\omega}_{\text{obj}}$	object angular velocity.
Robot state	
$\mathbf{q}^{\text{hand}}$	hand sub-vector of $\mathbf{q}$.
$\mathbf{q}^{\text{hand}}_{\text{def}}$	default hand pose.
$N_{\text{arm}}, N_{\text{hand}}$	arm (6) and hand (16) DOF counts.
l	arm-link (incl. palm) index.
z_l	base-frame height of link l.
h_l	clearance threshold, 0.1 mid-arm and 0.05 distal/palm.

B.5 Teacher and student training

Widths for both networks and all PPO and distillation hyperparameters are in Tables 7 and 8. Training uses rl_games [77]. All teacher PPO and student distillation runs use a single RTX 5090 GPU.

Distillation loss. The main-text objective in full, with the fixed BC weight κ written explicitly (it is absorbed into λ_D in the main text), is

$$\mathcal{L}_{\text{student}} \;=\; \mathcal{L}_{\text{PPO}}(\pi^{\text{S}}) \;+\; \lambda_D \cdot \kappa \cdot \mathbb{E}\big[\lVert\mu^{\text{S}}(o^{\text{S}}) - \mu^{\text{T}}(o^{\text{T}})\rVert_2^2\big], \tag{4}$$

where o^{S} and o^{T} are the paired onboard and privileged observations on the same transition, $\mathcal{L}_{\text{PPO}}$ aggregates the standard clipped surrogate, critic regression, entropy bonus, and action-bounds penalty, and κ is a fixed BC weight. λ_D is annealed linearly from 1.0 to 0.1 over the first 1,000 epochs. The wrist-depth image is additionally perturbed during distillation with the correlated depth-noise model [72].

B.6 Adaptive domain randomization and curriculum

Domain randomization combines *static DR*, sampled at reset from the fixed ranges in Table 9, and *adaptive DR*, which interpolates the parameters in Table 10 under a curriculum scalar $\rho \in [0, 1]$. Each environment holds a per-episode difficulty $d_i \in \{0, \ldots, 10\}$ that is promoted on trajectory success and demoted otherwise, and $\rho = \frac{1}{10 N_{\text{env}}} \sum_i d_i$ is the normalized population mean over the N_{env} parallel environments. Following DextrAH-RGB [10, 70], we bind the whole curriculum to this scalar but compute it from the per-environment population rather than a global success average.

Table 6: Reward terms. Phase letters G, M, R map to approach, in-hand+goal, and disengage.

Term	Weight	Phase	Expression
Tracking			
Lookahead tracking	$+8.0$	G, M	$\frac{1}{2}[e^{-e_p/0.03} + e^{-e_\theta/0.4}] \cdot c(\mathbf{f}) \cdot g_{\text{hp}}$
Trajectory success	$+25.0$	R	$0.1 \cdot \frac{1}{2}[e^{-e_p/0.03} + e^{-e_\theta/0.4}]$ $+ 0.9 \cdot \mathbb{1}[e_p < \tau_p \wedge e_\theta < \tau_\theta] \cdot g_{\text{cont}}(F_{\text{tot}}) \cdot g_{\text{hp}}$
Tracking progress	$+4.0$	all	$\text{clip}\left((e_{t-\Delta} - e_t)/e_{t-\Delta},\ 0, 1\right)$
Timeout bonus	$+200.0$	all	$\mathbb{1}$[natural episode end]
Palm-pose shaping			
Palm-pose following	$+3.5$	all	$\frac{1}{2}[e^{-e_p^h/\sigma_p} + e^{-e_\theta^h/\sigma_\theta}]$
Good finger contact	$+1.5$	M, R	$[\mathbb{1}_{\phi=R}(1 - c(\mathbf{f})) + \mathbb{1}_{\phi\neq R}\, c(\mathbf{f})] \cdot g_{\text{hp}}$
Finger manipulation	$+6.0$	M, R	$\frac{1}{2}[(1 - e^{-\|\Delta \mathbf{p}\|/0.015}) + (1 - e^{-\|\Delta\theta\|/0.15})] \cdot \mathbb{1}[e_t < e_{t-\Delta}] \cdot c(\mathbf{f}) \cdot g_{\text{hp}}$
Regularizers and safety			
Action L_2	-0.05	all	$\frac{1}{N_{\text{arm}}}\|\mathbf{a}_t^{\text{arm}}\|^2 + \frac{0.175}{N_{\text{hand}}}\left(\|\mathbf{a}_t^{\text{eig}}\|^2 + \|\mathbf{a}_t^{\text{hnd}}\|^2\right)$
Finger regularizer	-1.5	G, R	$1 - \exp\big(-\sqrt{\frac{1}{N_{\text{hand}}}\|\mathbf{q}^{\text{hand}} - \mathbf{q}_{\text{def}}^{\text{hand}}\|^2}\,/\,0.4\big)$
Object stillness	-0.3	R	$\frac{1}{2}[\tanh(\|\mathbf{v}_{\text{obj}}\|/0.02) + \tanh(\|\boldsymbol{\omega}_{\text{obj}}\|/0.2)]$
Arm-table penalty	-1.0	all	$\text{mean}_l\, \mathbb{1}[z_l < 0.255 + h_l]$
Finger self-contact	-0.1	all	$\text{mean}_{b\in\mathcal{B}_{\text{sc}}}\, \|\mathbf{f}_b^{\text{net}} - \mathbf{f}_b^{\text{obj}}\|$
Early termination	-100.0	all	$\mathbb{1}$[safety termination]

Table 7: Teacher and student network architectures. Widths are per-block.

Component	Configuration
Teacher π^T	
Token hidden dim	64
Attention heads	4 (head dim 16)
Point encoder	per-point MLP $[64, 64]$, ELU
Proprio encoder (query)	MLP $[64, 64]$, ELU
Cross-attention (query $\rightarrow$ points)	1 block
Shared trunk	MLP $[512, 256, 128]$, ELU
Heads	linear $\mu \in \mathbb{R}^{27}$, $\log\sigma \in \mathbb{R}^{27}$, $V \in \mathbb{R}$
Student π^S	
Token hidden dim	128
Attention heads	8 (head dim 16)
Wrist-depth tokenizer	4 stride-2 conv stages, channels $[32, 64, 128, 128]$, BN+ReLU; flattened to patch tokens of dim 128
Point encoder	per-point MLP $[128, 128]$, ELU
Proprio encoder (query)	MLP $[128, 128]$, ELU (proprioception only)
Cross-attention (query $\rightarrow$ point $\cup$ depth tokens)	1 block
Shared trunk	MLP $[1024, 512, 256, 128]$, ELU
Heads	linear $\mu \in \mathbb{R}^{27}$, $\log\sigma \in \mathbb{R}^{27}$, $V \in \mathbb{R}$

B.7 Real-to-sim calibration

Several measures complement the parametric randomization of Table 9 and the ADR schedule of Table 10.

- **Actuator system identification.** Per-joint stiffness, damping, and joint friction on the UR5e arm and LEAP hand are identified with CMA-ES [78] against recorded step responses; the fit sets the nominal actuator values that Table 9 perturbs.
- **Depth rendering and noise.** Wrist and external depth streams are rendered in Isaac Lab at the native resolutions of the deployment sensors and perturbed at observation time with the correlated depth-noise model of Handa et al. [72].

Table 8: Teacher PPO and student distillation hyperparameters.

Hyperparameter	Value
Teacher PPO	
Parallel environments	20,480
Horizon (steps / env / rollout)	32
Minibatch size	32,768
Mini-epochs per update	4
Optimizer	Adam
Learning rate	5×10^{-4} (adaptive KL)
KL threshold	0.016
Discount γ	0.998
GAE λ	0.95
Entropy coefficient	0.0025
Clip ratio	0.2
Critic coefficient	4.0
Gradient-norm clip	1.0
Student distillation	
Parallel environments	2,048
Horizon (steps / env / rollout)	16
Minibatch size	8,192
Mini-epochs per update	4
Learning rate	3×10^{-4} (adaptive KL)
Entropy coefficient	0.002
BC coefficient κ	10.0
λ_D initial $\rightarrow$ floor	$1.0 \rightarrow 0.1$ (linear, 1,000 epochs)
Other settings	as teacher

Table 9: Static domain-randomization ranges.

Group	Parameter	Range
Object	per-instance scale	$[0.65, 1.1]$
	mass (scale of nominal)	$[0.1, 1.6]$
	friction (static / dynamic)	$[0.5, 1.0]$
Robot body	friction (static / dynamic)	$[0.5, 1.0]$
Robot actuators	stiffness (scale of nominal)	$[0.8, 1.2]$
	damping (scale of nominal)	$[0.8, 1.2]$
	joint-friction (scale of nominal)	$[0.8, 1.2]$
Reset-time jitter	robot base (x, y, z)	$[-0.01, 0.01]$ m
	wrist-camera forward offset	$[0.0, 0.002]$ m
	wrist-camera lateral/vertical offset	$[-0.002, 0.002]$ m
Student obs noise (σ)	joint positions (arm)	0.005 rad
	joint positions (hand)	0.05 rad
	eigen-grasp projection $\mathbf{c}_t$	0.01
	palm pose, palm-pose lookahead	0.005 each
	camera-visible query points, lookahead	0.005 each

- **Reset-time jitter.** At each episode reset we apply static jitter to the robot's joint positions, base pose, and wrist-camera mounting offset.
- **Student observation noise.** During distillation every student observation channel carries per-channel Gaussian noise at the fixed amplitudes listed at the bottom of Table 9.

B.8 Deployment pipeline

The sensorimotor policy π^{S} is ticked at the same 50 Hz as in simulation. Wrist-camera depth is produced by Fast-FoundationStereo [79] on the IR pair rather than the sensor's onboard depth stream, which we found too noisy at the close range that π^{S} conditions on (Fig. 10). Rates of every pipeline

Table 10: ADR-interpolated parameters. All sweep linearly from initial to final as $\rho : 0 \to 1$.

Parameter	**Initial** ($\rho = 0$)	**Final** ($\rho = 1$)
Gravity $\|\mathbf{g}\|$	0.0981 m/s^2	9.81 m/s^2
Trajectory-deviation termination (e_p, e_θ)	(0.20 m, 2.0 rad)	(0.10 m, 1.0 rad)
Palm-pose termination (e_p^h, e_θ^h)	(0.15 m, 2.0 rad)	(0.075 m, 0.9 rad)
Trajectory-success tolerance (τ_p, τ_θ)	(0.07 m, 0.7 rad)	(0.04 m, 0.4 rad)
Object wrench (force, torque) per axis	0, 0	± 0.15 N, ± 0.01 N m

Table 11: Deployment tick rates.

Stage	**Rate (Hz)**
SAM 3.1 mask refresh	1
Intent model f_θ	1
DenseTrack3Dv2 sliding-window tracking	30
Sensorimotor policy π^{S}	50

stage are listed in Table 11. The entire deployment stack (SAM 3.1, intent model, point tracker, sensorimotor policy, Fast-FoundationStereo) runs on a single RTX 5090 GPU.

C Experimental Details

C.1 Baselines

Open-loop video-generation planner (§4.1). For each rollout (each generation takes 2–3 minutes), we generate a fresh video plan from the initial RGB observation, filtering out generations that depict random or clearly incorrect behavior and resampling until a plausible plan is obtained. Object flow and the palm-pose reference are extracted with the same pipeline used for human-video supervision (App. A.1), except that the entire generated video is processed as one continuous window rather than overlapping windows, so a single non-windowed flow and palm-pose trajectory is produced. See Fig. 11 for an example rollout.

Parallel-jaw gripper sensorimotor policy (§4.3). Trained in simulation with the same procedural trajectories, objects, and reward as the LEAP hand policy, with the *Finger manipulation* reward removed. The two gripper jaws stand in for the fingers in the remaining contact-gated terms: the contact factor $c(\mathbf{f})$ collapses to $c(\mathbf{f}) = 2\, s_{\text{jaw1}} + 2\, s_{\text{jaw2}}$, and *Good finger contact* uses this collapsed form. The intent model's palm-pose reference, supervised against a human palm, is mapped to the gripper by aligning the palm frame so that the gripper's closing axis matches the four-fingers-to-thumb opposition of the human hand.

C.2 Per-scenario failure-mode breakdown

We score each rollout as one of four outcomes; the legend colors match Fig. 12 and Fig. 13.

- **Success**: the rollout completes the task (per-task criteria are defined in §4.1 and §4.3).
- **Unrecoverable state**: the rollout enters a configuration from which the policy cannot make progress. Common cases include the manipulated object leaving the workspace (e.g., the apple rolling off the table in binning), or the robot getting stuck in a state where it cannot feasibly continue (e.g., the spoon is grasped at an orientation from which completing the rotations would push the wrist past its joint limit). Once this state is reached, no further closed-loop intent update can recover the rollout, and we stop the trial.
- **Perception loss**: the upstream perception pipeline loses the object during the rollout. Typical causes are the robot's hand or arm fully occluding the object for long enough that SAM 3.1's mask drifts off the object or DenseTrack3Dv2's tracked points reproject onto

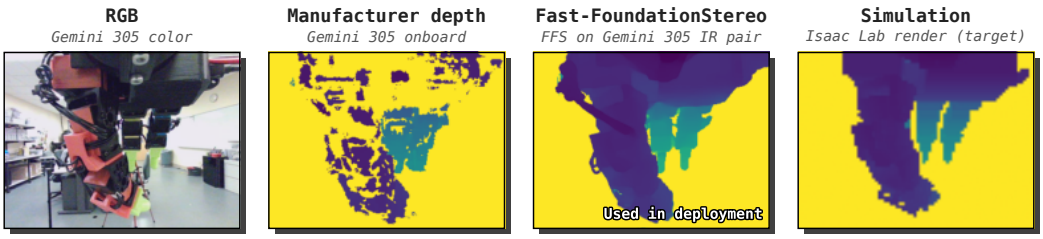

Figure 10: **Wrist-camera depth.** RGB and three depth streams from the wrist-mounted Gemini 305: the manufacturer's onboard depth, Fast-FoundationStereo [79] on the IR pair (5 ms inference), and the Isaac Lab simulation depth used at training. `LUCID` deploys with the FFS stream.

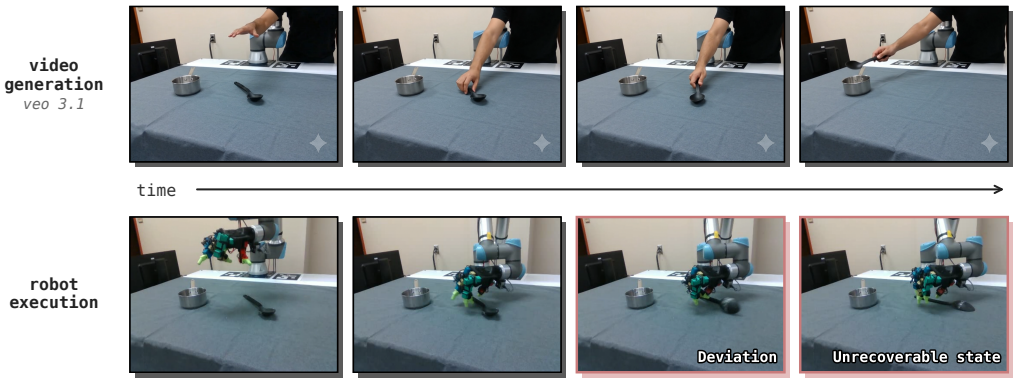

Figure 11: **Open-loop video-generation planner.** Veo 3.1 generates a human video plan from the initial scene; object flow and palm pose are extracted and executed by the sensorimotor policy. The plan is fixed, so execution can diverge.

the wrong surface. Without a valid mask or 3D track, the intent model has no query to predict from and the policy stalls.

- **Incorrect behavior**: the rollout neither succeeds nor enters a hard failure, but the policy executes a plan that does not actually complete the task. For example, in wiping, the marker remains visible on the whiteboard but the policy lifts the cloth and disengages anyway; or in binning, the predicted object flow points away from the target container, so the policy carries the object to the wrong location, often outside the camera bounds.

C.3 Intent-Loss Scaling Extrapolation

To visualize extrapolation uncertainty, we fit fixed-exponent power laws $L(M) = c + aM^{-\alpha}$ to the five held-out intent-loss points from Fig. 6, where M is the number of training clips. For each $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$, c and a are fit by least squares.

The candidates are nearly indistinguishable over the observed 1K–20K clip range. The right panel extends the same fits $1000\times$ beyond the largest training set. Thus the current sweep supports the trend that more video helps but does not pin down a precise long-range scaling forecast.

C.4 Query-points ablation

We sweep the number of camera-visible query points N the student policy receives during training (Fig. 15). Episode reward continues to rise with more points but with diminishing returns. We pick $N = 16$ as our operating point: it captures most of the reward gain while keeping training fast for the intent model.

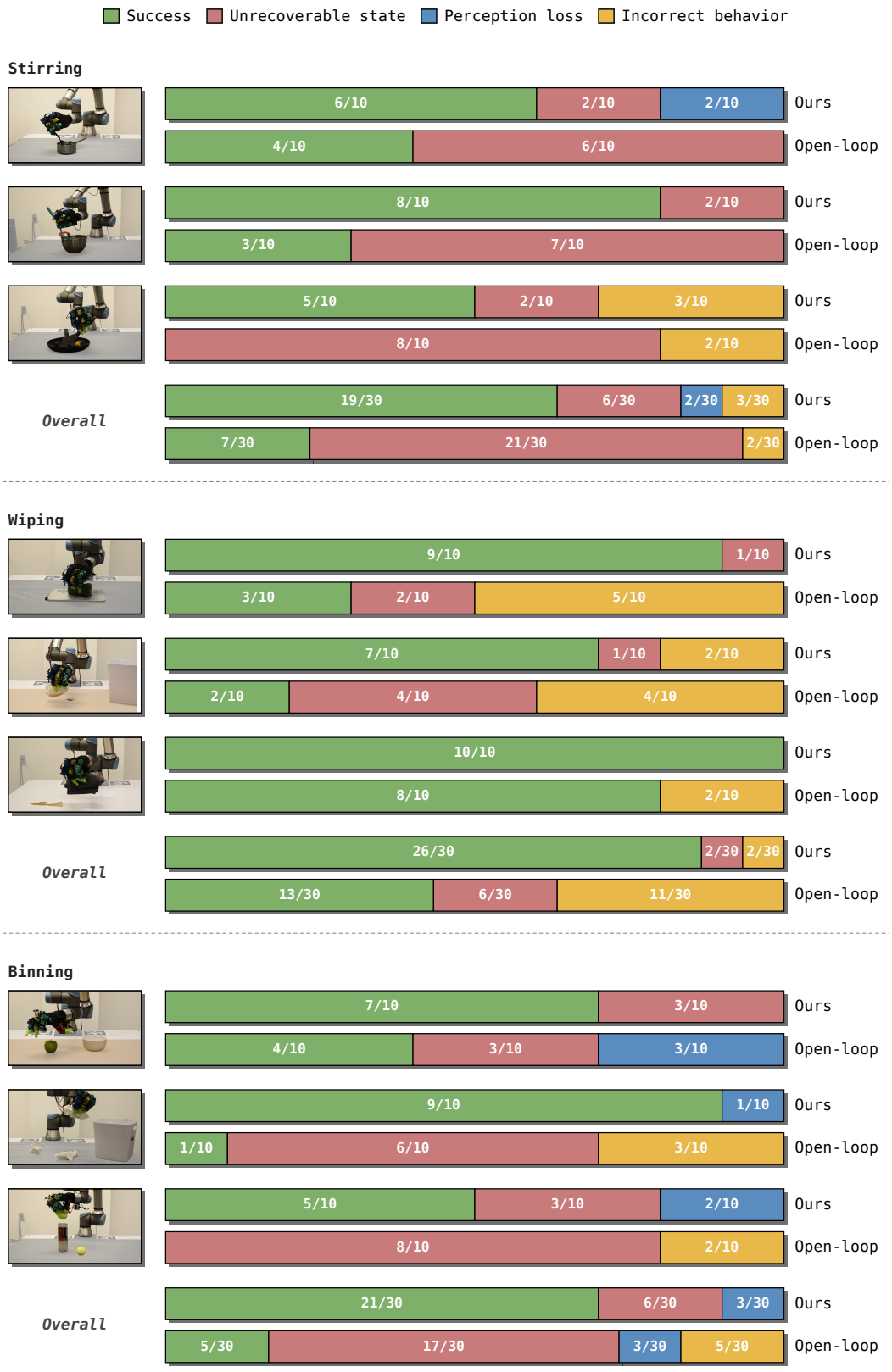

Figure 12: **Failure-mode breakdown: web-scale tasks.** Per-trial outcomes for Stirring, Wiping, and Binning across the three evaluation scenarios from §4.1.

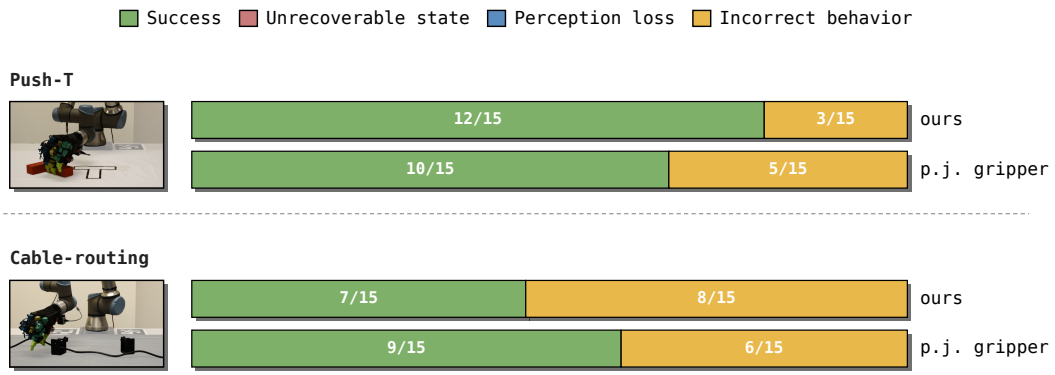

Figure 13: **Failure-mode breakdown: self-collected tasks.** Per-trial outcomes for Push-T and Cable-routing. Each row compares execution with the dexterous hand policy and the parallel-jaw gripper policy.

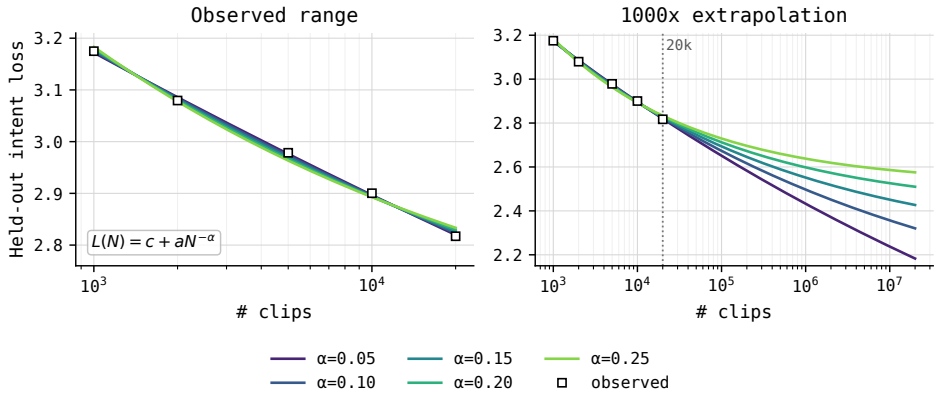

Figure 14: **Power-law extrapolations of intent loss.** Fixed-exponent fits to the held-out intent-loss points from Fig. 6. The measured range supports many similar fits, while extrapolation reveals substantial uncertainty in the long-range forecast.

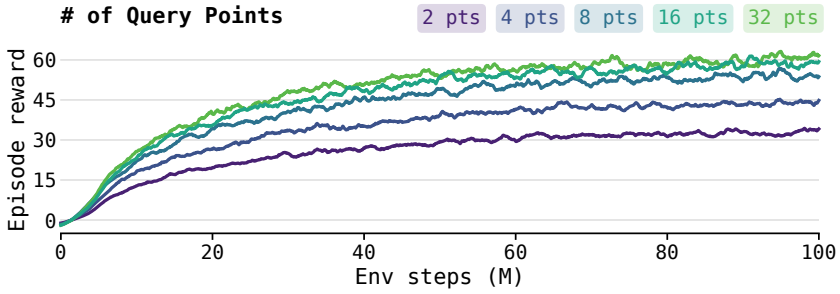

Figure 15: **Query-points ablation.** Episode reward vs environment steps as we sweep the number of camera-visible query points N the student policy receives.