CARAT: STATIC COMPATIBILITY AND DYNAMIC READINESS FOR ADAPTIVE HUMAN-TO-ROBOT DEXTEROUS MANIPULATION TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferring human demonstrations offers a promising solution to the scarcity of embodiment-specific robot data in dexterous manipulation learning. However, existing human-to-robot transfer methods leave two complementary sources of variation underexplored: the reusability of human interaction priors differs across task-hand pairs, while the learner's competence varies across trajectory phases and evolves throughout training. To exploit these variations, we propose CARAT, a unified framework that leverages Static **C**ompatibility **A**nd Dynamic **R**eadiness to **A**daptively facilitate human-to-robot manipulation **T**ransfer. CARAT formulates a Static Compatibility metric to calibrate task-hand-specific interaction constraints, balancing human prior preservation with alternative contact discovery. It also employs a Dynamic Readiness scheduler to track phase-wise policy competence online, adaptively shifting training emphasis toward the current learning frontier. Across seven manipulation tasks and four heterogeneous dexterous hands, CARAT outperforms baselines under a unified evaluation protocol, demonstrating the vital role of exploiting these variations in human-to-robot transfer. The source code will be released upon paper acceptance.

1 INTRODUCTION

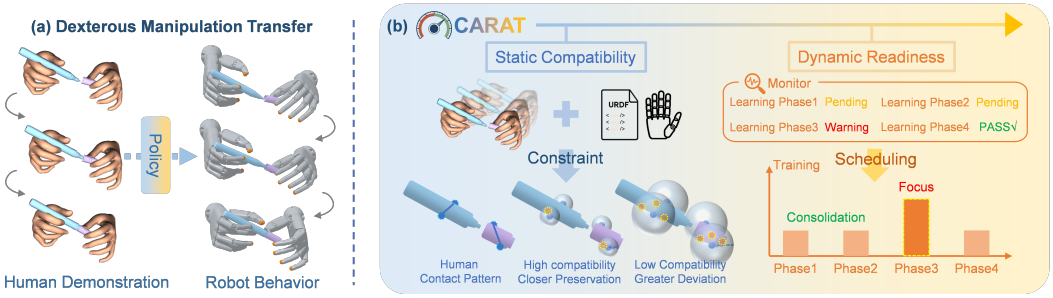


Figure 1: **CARAT for adaptive human-to-robot dexterous manipulation transfer.** (a) Human demonstrations are converted into executable robot behaviors through policy learning. (b) CARAT combines Static Compatibility for interaction calibration and Dynamic Readiness for adaptive training allocation.

Learning-based approaches have achieved substantial progress in dexterous manipulation (Rajeswaran et al., 2018; Andrychowicz et al., 2020; Qin et al., 2023a; Li et al., 2023; Zhang et al., 2025; 2024; Luo et al., 2026). Yet scaling them remains heavily constrained by the scarcity of high-quality robot interaction data (Wang et al., 2024; Li et al., 2025). Collecting embodiment-specific demonstrations typically relies on labor-intensive teleoperation (Wang et al., 2024; Iyer et al., 2025; Cheng et al., 2025). A promising alternative is translating readily available human hand-object demonstrations into robot-object interaction data for target embodiments (Fig. 1, left) (Qin et al., 2022; Li et al., 2025; Lum et al., 2025; Zhou et al., 2026; Lou et al., 2026). This bypasses real-time teleoperation and enables rapid data generation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Recent human-to-robot manipulation transfer methods commonly first construct embodiment-specific references from human interactions, and then use policy learning to turn these references into executable robot behaviors that reproduce the demonstrated object behavior (Zhao et al., 2026; Li et al., 2025; Liu et al., 2025). The effectiveness of this paradigm shows that human demonstrations provide useful priors for robot policy learning, motivating a closer examination of what further information in the transfer process can guide learning. In practice, the same human demonstration can induce different feasible interaction strategies across robot hands (Xu et al., 2025; Zhao et al., 2026), and different trajectory phases present varying challenges to the learner (Xu et al., 2026).

We explicitly characterize these variations along two complementary axes. First, the reusability of a human interaction prior is task-hand specific. Although similarity between human and robot hands aids transfer, different tasks impose varying requirements on contact precision and motion complexity. Hence, prior reusability should be characterized at the level of the task-hand pair. Second, learner competence is phase dependent and evolves throughout training. Different trajectory phases impose different interaction demands, whose effective difficulty further depends on the target embodiment and the current policy. Training should therefore adapt to both what interaction prior can be reused and what the learner is currently ready to master.

Based on this view, we propose CARAT, a unified framework that incorporates these variations into the reinforcement learning pipeline for learning transfer policies. First, Static Compatibility estimates prior reusability for a given task-hand pair to shape the reward landscape. It calibrates the spatial tolerance of soft interaction constraint rewards: high compatibility tightens the constraint to preserve demonstrated local interaction patterns, whereas low compatibility relaxes it to encourage discovering alternative embodiment-specific contacts (Fig. 1 blue). Second, Dynamic Readiness acts as an online monitor that tracks phase-wise policy competence. It identifies the current learning frontier, which is the trajectory phase not yet mastered but closest to the policy's capability boundary. Guided by this readiness, a reset-mixture scheduler then increases the probability of initializing training near this frontier while maintaining coverage of the full trajectory (Fig. 1 orange). In this way, CARAT utilizes Static Compatibility to determine how strongly the human prior should be preserved, and Dynamic Readiness to determine where training effort should be concentrated. Experiments show that CARAT outperforms the baselines under a unified evaluation protocol, demonstrating the benefit of jointly accounting for these variations.

Our contributions are summarized as follows:

• We introduce Static Compatibility, a task-hand-specific metric that shapes the reward function by calibrating an interaction tolerance, balancing prior preservation and alternative contact discovery.

• We introduce Dynamic Readiness, an online estimate of phase-wise policy competence that identifies the current learning frontier. Guided by this estimate, a reset-mixture scheduler reallocates training emphasis while preserving full-trajectory coverage.

• We evaluate CARAT on 7 manipulation tasks and 4 heterogeneous dexterous hands. CARAT achieves the best average strict success rate under a unified protocol.

2 Related Work

2.1 Kinematic Hand Motion Retargeting

Kinematic hand-motion retargeting maps human motions to robot joint configurations, primarily preserving geometric correspondence across embodiments (Handa et al., 2020; Qin et al., 2023b; Junge & Hughes, 2025; Yin et al., 2025). Such mappings support real-time teleoperation and demonstration collection (Qin et al., 2023b; Iyer et al., 2025; Cheng et al., 2025; Lai et al., 2026a). However, contact-rich teleoperation keeps the human in the control loop, making data collection costly and embodiment-dependent. This motivates offline human-to-robot transfer from prerecorded human hand-object interactions, avoiding repeated human-in-the-loop demonstration collection for each target embodiment. Accordingly, our work adopts this offline setting and focuses on how prerecorded human priors can be converted effectively into suitable robot references.

2.2 HUMAN-TO-ROBOT DEXTEROUS MANIPULATION TRANSFER

Beyond kinematic hand-motion retargeting, human-to-robot dexterous manipulation transfer aims to produce robot-executable interaction trajectories and policies from human demonstrations (Li et al., 2025; Zhao et al., 2026; Zhu et al., 2026). Broadly, existing approaches emphasize two non-exclusive directions: constructing higher-quality robot references and learning policies that resolve the remaining embodiment and contact-dynamics mismatch. One line improves robot references through geometric and contact-aware optimization (Lin et al., 2025; Wu et al., 2026; Lai et al., 2026b), or physics-aware optimization and retargeting (Liu et al., 2024; Pan et al., 2025; Paliwal et al., 2026). A complementary line uses human-derived interaction priors to guide robot policy learning (Zhou et al., 2026; Liu et al., 2025; Xu et al., 2025; Yuan et al., 2025; Feng et al., 2026).

Within reference-guided policy learning, ManipTrans uses residual adaptation, while DexMachina eases exploration through decaying virtual-object assistance (Li et al., 2025; Zhao et al., 2026). CHORD introduces object-centric contact-wrench guidance, whereas Dexplore treats demonstrations as soft guidance through adaptive spatial scopes (Zhu et al., 2026; Xu et al., 2025). DemoBot and ConTrack further support long-horizon learning through segment-aware and policy-reachable mid-trajectory resets, respectively (Xu et al., 2026; Liang et al., 2026). Taken together, existing methods improve different components of transfer, but largely treat the reusability of human interaction priors and the learner's evolving phase-wise competence independently. CARAT explicitly models these two sources of variation: Static Compatibility estimates task-hand-specific prior reusability offline to calibrate interaction tolerance, while Dynamic Readiness tracks phase-wise competence online to adapt training toward the current learning frontier.

3 METHOD

3.1 PROBLEM FORMULATION

We study human-to-robot dexterous manipulation transfer from prerecorded human hand-object demonstrations in simulation. A demonstration is represented as $\tau_h = \{(\{h_t^s\}_{s\in\mathcal{S}}, g_t)\}_{t=1}^{T}$, where $\mathcal{S} \subseteq \{\mathrm{L}, \mathrm{R}\}$ denotes the participating hand sides, h_t^s is the human hand state, and g_t denotes the poses of the tracked object parts. Given a target robot embodiment e, a reference-construction pipeline G maps the human demonstration τ_h to an initial robot reference trajectory $\tau_r^0 = G(\tau_h, e) = \{r_t^0\}_{t=1}^{T}$. At each timestep, a policy π predicts an action conditioned on the current robot-object observation o_t^e and the time-aligned robot and object references (r_t^0, g_t): $a_t \sim \pi\big(\cdot \mid o_t^e, r_t^0, g_t\big)$. Executing these actions under the dynamics and actuation constraints of embodiment e produces an object trajectory $\{\hat{g}_t\}_{t=1}^{T}$. The goal is to reproduce the demonstrated object evolution $\{g_t\}_{t=1}^{T}$ while maintaining dynamically feasible robot behavior.

3.2 OVERVIEW

CARAT builds on the reference-guided policy-learning formulation above and augments it with two complementary mechanisms (Fig. 2). Given the initial robot reference τ_r^0 constructed by G, Static Compatibility is computed before policy learning to estimate task-hand-specific prior reusability and calibrate the tolerance of an object-centric Soft Interaction Constraint. During policy learning, Dynamic Readiness continuously tracks the learner's phase-wise competence and identifies the current learning frontier. Then Reset-Mixture Scheduler places greater training emphasis near this frontier while preserving coverage of the full trajectory.

3.3 REFERENCE CONSTRUCTION

To isolate the contribution of CARAT, all policy-learning methods share the same reference construction pipeline G, implemented following ManipTrans (Li et al., 2025). Specifically, optimization-based kinematic retargeting fits the robot wrist pose and joint configuration to the corresponding human hand landmarks under joint limits, providing robot-side initialization. A pre-trained and frozen trajectory imitator then produces the base actions, whose rollout defines τ_r^0. Fixed human-robot semantic landmark correspondences are adopted from ManipTrans.

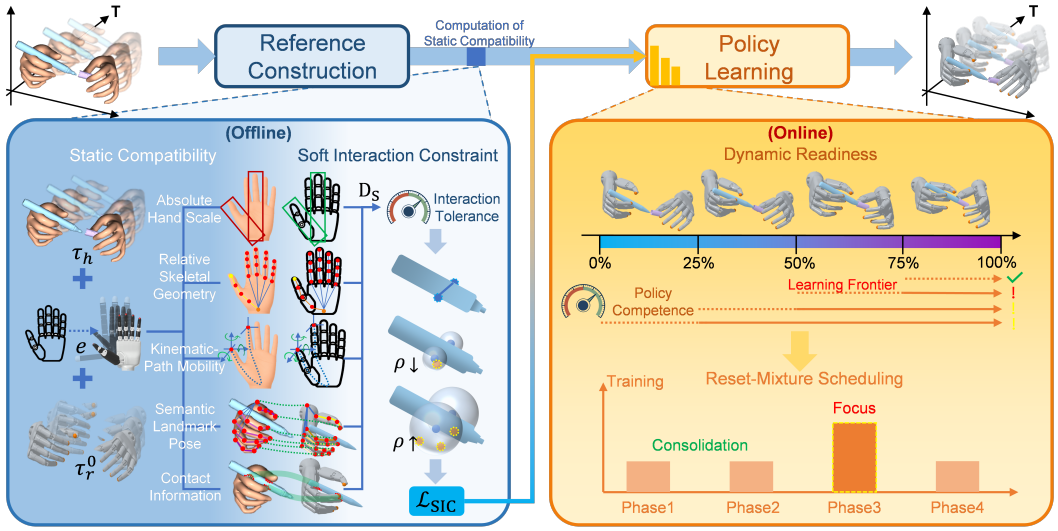


Figure 2: **Overview of CARAT architecture.** Static Compatibility is computed offline before policy learning to calibrate the tolerance radius of the Soft Interaction Constraint. Dynamic Readiness is updated online to adapt reset allocation toward the current learning frontier. Together, they determine how strongly the human prior is preserved and where training is concentrated.

3.4 Static Compatibility and Soft Interaction Constraint

Human interaction priors are not equally reusable across task-hand pairs. Static Compatibility estimates this reusability before policy learning and uses it to determine how faithfully the demonstrated interaction should constrain the target policy. We represent Static Compatibility through a static discrepancy $D_{\mathrm{S}} \in [0, 1]$, where a smaller value indicates a more reusable human prior. As illustrated in Fig. 2, D_{S} is estimated from both intrinsic embodiment mismatch and task-specific interaction mismatch, and subsequently converted into an object-centric interaction tolerance, calibrating how far the robot is allowed to deviate from the demonstrated interaction pattern.

3.4.1 Static Compatibility Estimation

Intrinsic Morphology Discrepancy. We first characterize whether the target embodiment provides the physical structure and motion capability required to realize the human interaction. Under fixed semantic landmark correspondences Π, three complementary factors are considered: absolute hand scale, relative skeletal geometry, and kinematic-path mobility:

$$D_{\mathrm{M}} = \mathrm{clip}\left(\omega_{\mathrm{size}} d_{\mathrm{size}} + \omega_L d_L + \omega_C d_C, 0, 1\right). \tag{1}$$

Here, d_{size} measures absolute-scale mismatch and d_L scale-invariant geodesic mismatch. For correspondences $x \leftrightarrow y$ and $x' \leftrightarrow y'$, we compare the mobility of robot path $y - y'$ and human path $x - x'$:

$$d_C = \sum_{x,x',y,y'} \Pi_{xy} \Pi_{x'y'} \left(\frac{[C_h(x, x') - C_r(y, y')]_+}{C_{\mathrm{ref}}} \right)^2, \tag{2}$$

where C_h, C_r are human and robot path-mobility descriptors, Π_{xy} weights landmark correspondence, $C_{\mathrm{ref}} > 0$ is a normalization constant, and $[u]_+ = \max(u, 0)$. Thus, d_C penalizes mobility deficits but not surplus robot mobility. Detailed definitions of the scale, geodesic, and actuation-aware mobility terms are provided in Appendix A.1, A.2 and A.3.

Task-Specific Interaction Discrepancy. Morphology alone does not determine whether a prior is reusable for a particular task. Therefore, we compare the demonstrated human interaction τ_h with the initial robot reference τ_r^0. Over aligned valid frames $\mathcal{V}$, we aggregate discrepancies in semantic landmark pose, contact location, contact normal, and contact state: $E_{\mathrm{int}} = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \left(E_{\mathrm{pose,t}} + E_{\mathrm{contact,t}} + E_{\mathrm{normal,t}} + E_{\mathrm{mask,t}} \right)$, which is normalized into $D_{\mathrm{I}} \in [0, 1]$. The per-term definitions and normalization are given in Appendix A.4.

Estimation of Static Discrepancy. Finally, the intrinsic and task-specific discrepancies are combined as:

$$D_{\mathrm{S}} = D_{\mathrm{M}} + \alpha(1 - D_{\mathrm{M}})D_{\mathrm{I}}, \qquad 0 \le \alpha \le 1. \tag{3}$$

This gated form treats intrinsic morphology as the baseline feasibility of reuse, while task-specific mismatch adjusts the remaining compatibility margin. Consequently, the same robot hand can receive different compatibility estimates for different manipulation tasks.

3.4.2 Compatibility-Calibrated Soft Interaction Constraint

Rather than directly using D_{S} as an additional policy input or reward weight, we parameterize how much deviation from the demonstrated interaction is admissible. Specifically, D_{S} is mapped to an interaction tolerance radius $\rho(D_{\mathrm{S}}) = \rho_{\min} + (\rho_{\max} - \rho_{\min})D_{\mathrm{S}}$, which is computed offline and fixed throughout policy learning; higher compatibility (lower D_{S}) yields a tighter tolerance, whereas lower compatibility permits greater adaptation.

To isolate interaction geometry from global object motion, we measure fingertip deviation in an object-centric interaction space. For a fixed active side s (side index omitted for clarity), let $j^h_{t,i}$ denote the i-th demonstrated human fingertip and $y^e_{t,i}$ its corresponding robot fingertip at the current rollout state. Let $n(i)$ denote the object part associated with fingertip i, whose reference pose $(p_{t,n(i)}, R_{t,n(i)})$ is contained in g_t. Using the corresponding current pose $(\hat{p}_{t,n(i)}, \hat{R}_{t,n(i)})$ from $\hat{g}_t$, we transport the object-relative demonstrated fingertip location through the current object pose:

$$\tilde{j}_{t,i} = \hat{p}_{t,n(i)} + \hat{R}_{t,n(i)} R^{\top}_{t,n(i)} \big(j^h_{t,i} - p_{t,n(i)}\big), \qquad e_{t,i} = \big\| y^e_{t,i} - \tilde{j}_{t,i} \big\|_2 . \tag{4}$$

Defining the normalized excess $z_{t,i} = \left[\frac{e_{t,i}}{\rho} - 1\right]_+$, the side-wise Soft Interaction Constraint is $\mathcal{L}^{(s)}_{\mathrm{SIC},t} = \frac{\sum_i \tilde{a}_{t,i}\phi_\delta(z_{t,i})}{\max\left(\sum_i \tilde{a}_{t,i}, \varepsilon\right)}$. Here, ϕ_δ is a pseudo-Huber penalty, while $\tilde{a}_{t,i}$ activates the constraint only during demonstrated interaction phases. Normalization by ρ makes the penalty reflect relative tolerance violation across heterogeneous task-hand pairs. More implementation details are provided in Appendix A.5. The resulting objective augments the reference-guided baseline reward:

$$\mathcal{R}_{\mathrm{CARAT},t} = \mathcal{R}_{\mathrm{base},t} - \lambda \sum_{s \in \mathcal{S}} \mathcal{L}^{(s)}_{\mathrm{SIC},t}, \tag{5}$$

where $\mathcal{R}_{\mathrm{base},t}$ uses the ManipTrans hand-motion tracking, object tracking, and contact rewards (Li et al., 2025). No additional penalty is incurred inside the tolerance region. Consequently, highly compatible pairs are encouraged to preserve reusable human interaction patterns, whereas less compatible pairs retain greater freedom to discover embodiment-specific contacts.

3.5 Dynamic Readiness and Reset-Mixture Scheduling

For a given task-hand pair, learning difficulty varies across trajectory phases and evolves with the policy. As illustrated in Fig. 2, Dynamic Readiness tracks phase-wise competence online to identify the current learning frontier, which is the phase not yet reliably mastered but closest to the policy's current capability. The Reset-Mixture Scheduler then increases the probability of starting training from this frontier while continuing to sample the other trajectory phases.

3.5.1 Look-Ahead Competence Evaluation

During policy learning, gravity and friction remain fixed at their target simulation values. The base training pipeline follows a predefined schedule of tracking-error thresholds, which determine whether the robot and object remain sufficiently close to their time-aligned references and trigger episode failure when the deviation becomes excessive. The thresholds are relaxed early in training to avoid premature failures when tracking is still inaccurate, and are progressively tightened according to the predefined schedule to enforce more precise reference following. Consequently, success under the current relaxed thresholds may overestimate whether the policy is ready for the upcoming training stage. We therefore additionally evaluate each rollout using the tighter thresholds scheduled for the next training interval, producing a binary look-ahead success label ξ^+_n. This evaluation reuses recorded rollout statistics and requires no additional simulation. Further details are provided in Appendix B.1.

3.5.2 DYNAMIC READINESS ESTIMATION

Each demonstration is divided into $B = 4$ equal temporal phases. When initialized from phase b, the policy must complete the remaining trajectory suffix from b to the end. Let $\mathcal{W}_b$ denote recent completed episodes initialized from phase b. The recent look-ahead completion rate is estimated as $\hat{C}_b = \frac{1}{|\mathcal{W}_b|} \sum_{n \in \mathcal{W}_b} \xi_n^+$. To reduce sensitivity to sparse or noisy estimates, we use the one-sided Wilson lower confidence bound $\underline{C}_b$ of $\hat{C}_b$ and define the phase-wise Dynamic Readiness

$$A_b = \mathrm{clip}\left(\frac{\underline{C}_b}{c_{\mathrm{front}}}, 0, 1\right), \tag{6}$$

where c_{front} denotes the minimum competence required to progress from phase b. $A_b = 1$ indicates sufficient competence to advance, rather than full mastery. Details are provided in Appendix B.2.

3.5.3 READINESS-GUIDED RESET-MIXTURE SCHEDULING

Phase b is a frontier candidate when $\underline{C}_b < c_{\mathrm{front}} \leq \underline{C}_{b+1}$, indicating that the policy is not yet competent from phase b, while it can reliably progress from the immediately subsequent phase $b+1$. Among valid candidates, the current learning frontier is defined as the phase with the sharpest local competence transition:

$$b^* = \arg\max_b [A_{b+1} - A_b]_+. \tag{7}$$

Let $p_{\mathrm{base}}(b)$ denote the baseline reference state initialization (RSI) distribution, which samples reset states along the reference trajectory rather than always starting from its beginning. We retain this distribution as the coverage component and allocate additional probability mass to the current frontier:

$$p_{\mathrm{train}}(b) = (1-\beta)p_{\mathrm{base}}(b) + \beta \mathbb{I}[b = b^*], \qquad 0 \leq \beta < 1. \tag{8}$$

The RSI component retains baseline trajectory coverage, while the additional component concentrates training near the current capability boundary. Frontier injection requires sufficient phase evidence and an unmet competence gate; otherwise, p_{base} is used. Scheduling details are provided in Appendix B.3. Dynamic Readiness therefore adapts reset allocation as policy competence evolves.

4 EXPERIMENTS

We first describe the experimental setup (Sec. 4.1), then evaluate Static Compatibility (Sec. 4.2), Dynamic Readiness (Sec. 4.3) and the complete CARAT framework (Sec. 4.4).

4.1 EXPERIMENTAL SETUP

Tasks and Embodiments. We select seven manipulation tasks from OakInkV2 (Zhan et al., 2024): *Cap the Pen*, *Put off Alcohol Lamp*, *Unscrew*, *Uncap the Jar*, *Rearrange*, *Place Inside*, and *Brush Whiteboard*. They span single- and bimanual manipulation, short- and long-horizon motion, and varying contact complexity. Four dexterous hands are adopted from ManipTrans (Li et al., 2025) (*Inspire*, *Allegro*, *Shadow*, and *Artimano*), yielding $7 \times 4 = 28$ task-hand pairs.

Training. CARAT is built on ManipTrans in Isaac Gym and trained with PPO using 4096 parallel environments at $1/60\,\mathrm{s}$. Unless otherwise specified, we follow the ManipTrans policy architecture and optimization settings. Methods within each comparison share the same simulator, optimization configuration, and training seeds. Additional details are provided in Appendix C.1.

Baselines. We compare against ManipTrans and DexMachina (Zhao et al., 2026). For fair comparison, DexMachina's virtual force curriculum is adapted to the ManipTrans backbone (DexMachina*), aligning control modalities and base rewards. Details are provided in Appendix C.2.

Evaluation Metrics. Frozen policies are evaluated deterministically under target physical parameters and final tracking thresholds, with all curricula disabled. Each checkpoint uses a fixed bank of 256 reference start frames. A strict success requires reaching the reference end without hand/object-tracking, contact, or numerical-safety violations after the initialization grace period. We report Strict AUC_{1000} (S-AUC_{1000}), the normalized area under the strict-success curve evaluated at epochs $100, 200, \ldots, 1000$, to measure learning efficiency, and Strict Epoch-1000 Success Rate (S-SR_{E1000}) to measure final performance. Full protocol details are provided in Appendix C.3.

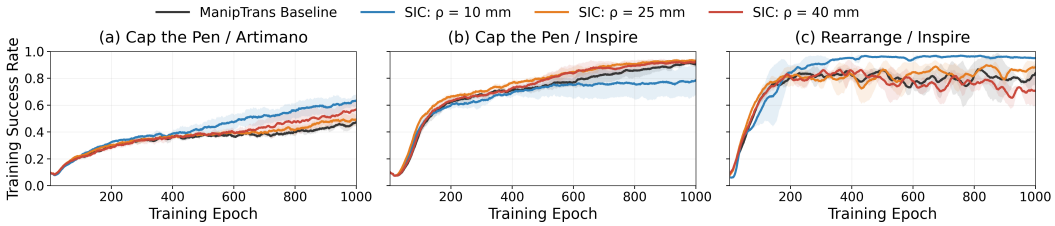


Figure 3: **Sensitivity to the interaction tolerance ρ on representative task-hand pairs.** Curves show three-seed mean **online success** (± 1 SD), smoothed over 20 epochs.

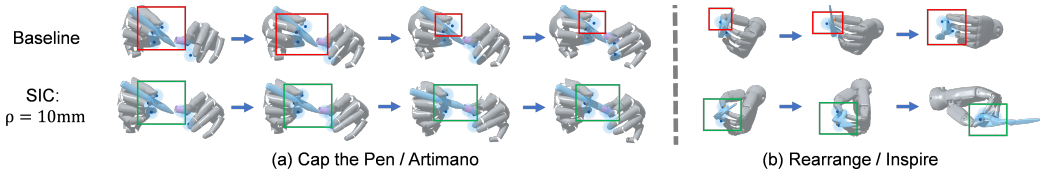


Figure 4: **Qualitative effect of interaction behaviors with and without SIC.** Compared with ManipTrans baseline, SIC with $\rho = 10\,\mathrm{mm}$ keeps the robot hand closer to the demonstrated object-relative interaction pattern, reducing interaction drift and enabling more stable task completion.

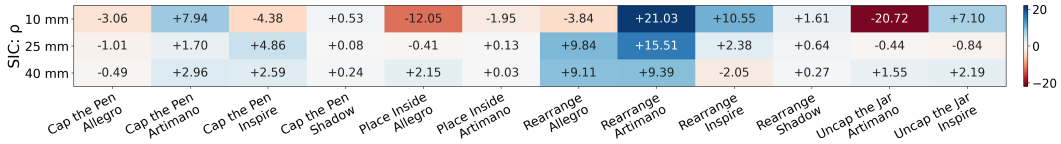


Figure 5: **Fixed-ρ sensitivity across task-hand pairs.** Each cell reports the change in **Online** AUC_{1000} (pp) relative to ManipTrans baseline, averaged over three seeds.

Table 1: **Strict comparison of interaction-tolerance variants.** Results are averaged over four hands and three seeds. In each task cell, left / right entries denote S-AUC_{1000} / S-SR_{E1000}.

S-AUC_{1000}(%) S-SR_{E1000}(%)	Cap the Pen (BiH)	Put off Alcohol Lamp (BiH)	Unscrew (BiH)	Uncap the Jar (BiH)	Rearrange (LH)	Place Inside (LH)	Brush Whiteboard (RH)	Average
Baseline	29.6 ± 1.8, 30.9 ± 1.2	44.9 ± 2.0, 56.8 ± 6.5	16.1 ± 1.3, 16.3 ± 2.8	27.3 ± 2.8, 38.9 ± 5.3	30.9 ± 2.1, 24.8 ± 3.8	56.9 ± 1.0, 59.1 ± 1.1	48.5 ± 2.2, 60.3 ± 0.4	36.3, 41.0
Fixed-$\rho = 25$ mm	35.2 ± 1.8, 36.5 ± 2.3	39.6 ± 5.5, 52.9 ± 5.9	17.1 ± 0.9, 18.3 ± 1.4	24.6 ± 0.8, 34.4 ± 3.2	32.1 ± 2.3, 34.3 ± 5.6	58.0 ± 0.2, 60.3 ± 1.5	51.1 ± 1.4, 56.2 ± 7.1	36.8, 41.8
SC-SIC	33.4 ± 4.9, 36.9 ± 6.6	43.1 ± 2.0, 49.3 ± 2.9	17.7 ± 0.4, 17.7 ± 1.3	29.2 ± 1.2, 41.4 ± 0.4	34.0 ± 5.3, 31.9 ± 4.6	58.9 ± 0.8, 60.9 ± 0.3	51.2 ± 0.8, 58.4 ± 3.7	38.2, 42.4

4.2 Evaluating Static Compatibility and Soft Interaction Constraint

We first test whether a single Soft Interaction Constraint (SIC) tolerance is appropriate for all task-hand pairs. To isolate the effect of the tolerance radius ρ, the ManipTrans baseline is augmented with the same SIC penalty $\mathcal{L}_{\mathrm{SIC},t}$ while fixing $\rho \in \{10, 25, 40\}\,\mathrm{mm}$, with all other training settings unchanged. Fig. 3 shows representative learning curves. For *Cap the Pen*, *Artimano* and *Inspire* favor different radii, whereas for the same *Inspire* hand, *Cap the Pen* and *Rearrange* exhibit different tolerance preferences. Fig. 4 visualizes the corresponding manipulation rollouts, comparing ManipTrans with the preferred fixed-radius SIC variant for each representative pair. These examples suggest that the appropriate interaction tolerance is specific to the task-hand pair. The pairs examined in Fig. 5 constitute our predefined calibration subset, on which the broader sweep further shows that no single radius consistently dominates across pairs. Having established this variation, we next test whether Static Compatibility can calibrate the interaction tolerance accordingly. Using only the predefined calibration subset, we select the fusion coefficient (Eq. (3)) $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ according to mean Online AUC_{1000}, yielding $\alpha = 0.7$. The Fixed-ρ baseline is independently tuned over $\{10, 25, 40\}\,\mathrm{mm}$ using the same criterion, yielding $\rho_{\mathrm{fixed}} = 25\,\mathrm{mm}$. Both hyperparameters are then frozen; strict evaluation metrics are not used for hyperparameter selection. Additional calibration details are provided in Appendix D.

Table 1 compares ManipTrans baseline, Fixed-ρ, and the Static Compatibility-calibrated SIC (SC-SIC) under the strict evaluation protocol. SC-SIC achieves the best average performance on both metrics, improving S-AUC$_{1000}$ from 36.3% (ManipTrans) and 36.8% (Fixed-ρ) to 38.2%, while also attaining the highest average S-SR$_{E1000}$ (42.4%). These results support SC-SIC as a more effective alternative to a single globally fixed interaction tolerance. Full learning curves for all task-hand pairs are provided in Appendix G.

4.3 Evaluating Dynamic Readiness and Reset-Mixture Scheduling

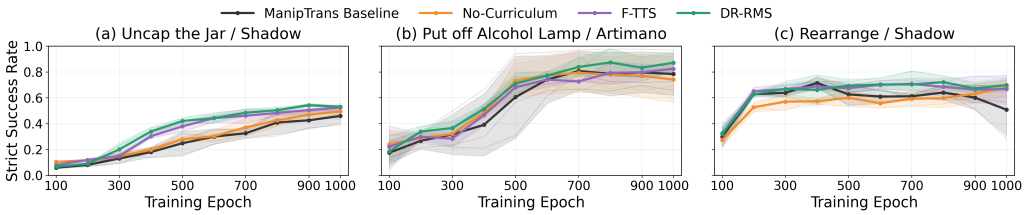


Figure 6: **Curriculum ablation on representative task-hand pairs.** Thick lines show three-seed means, bands ± 1 SD, and thin lines individual seeds.

Table 2: **Strict comparison of curriculum variants.** Results are averaged over four hands and three seeds. In each task cell, left / right entries denote S-AUC$_{1000}$ / S-SR$_{E1000}$.

S-AUC$_{1000}$(%) S-SR$_{E1000}$(%)	Cap the Pen (BiH)	Put off Alcohol Lamp (BiH)	Unscrew (BiH)	Uncap the Jar (BiH)	Rearrange (LH)	Place Inside (LH)	Brush Whiteboard (RH)	Average
Baseline	29.6 ± 1.8, 30.9 ± 1.2	44.9 ± 2.0, 56.8 ± 6.5	16.1 ± 1.3, 16.3 ± 2.8	27.3 ± 2.8, 38.9 ± 5.3	30.9 ± 2.1, 24.8 ± 3.8	56.9 ± 1.0, 59.1 ± 1.1	48.5 ± 2.2, 60.3 ± 0.4	36.3, 41.0
DR-RMS	32.0 ± 3.7, 37.3 ± 9.7	47.3 ± 2.7, 60.1 ± 6.0	16.9 ± 0.8, 18.5 ± 0.8	32.6 ± 0.7, 46.5 ± 2.3	38.7 ± 3.2, 39.8 ± 2.9	57.6 ± 0.1, 59.0 ± 1.1	42.3 ± 5.2, 51.7 ± 5.5	38.2, 44.7

To examine the effect of Dynamic Readiness and Reset-Mixture Scheduling (DR-RMS), we conduct a controlled ablation on three representative task-hand pairs. All variants use the same ManipTrans architecture and reward, and differ only in curriculum configuration. We compare the ManipTrans baseline, training without curriculum, a predefined Fixed Tracking-Threshold Schedule under target physics (F-TTS), and F-TTS augmented with DR-RMS. Since F-TTS and DR-RMS differ only in reset allocation, their comparison directly isolates the contribution of Dynamic Readiness. Fig. 6 shows the strict learning curves for the three representative pairs. F-TTS already improves training over the weaker curriculum settings, while adding DR-RMS further produces more favorable learning trends. We then extend the evaluation to all 28 task-hand pairs (Table 2). DR-RMS improves S-AUC$_{1000}$ from 36.3% to 38.2% and S-SR$_{E1000}$ from 41.0% to 44.7%. This broader improvement indicates that readiness-guided reset allocation generalizes beyond the representative cases in Fig. 6. Full task-wise curriculum results, visualizations of the detected learning frontier, and learning curves for all task-hand pairs are provided in Appendix E, F and G.

4.4 Overall Comparison

We finally evaluate the complete CARAT framework across all seven tasks and four hand embodiments. Fig. 7 compares CARAT with ManipTrans and DexMachina* over the resulting 28 task-hand pairs (learning curves provided in Appendix G). While performance varies across individual pairs, CARAT achieves the strongest average performance across the heterogeneous tasks and embodiments. Table 3 summarizes the two baselines, the individual CARAT components, and the complete framework. Compared with ManipTrans, CARAT improves S-AUC$_{1000}$ from 36.3% to 40.6% and S-SR$_{E1000}$ from 41.0% to 46.7%, corresponding to gains of 4.3 and 5.7 percentage points. Both SC-SIC and DR-RMS individually improve over ManipTrans, while their combi-

Table 3: **Average strict performance over 28 task-hand pairs.**

	S-AUC$_{1000}$ (%)	S-SR$_{E1000}$ (%)
DexMachina*	28.2	33.6
ManipTrans	36.3	41.0
SC-SIC	38.2	42.4
DR-RMS	38.2	44.7
CARAT	**40.6**	**46.7**

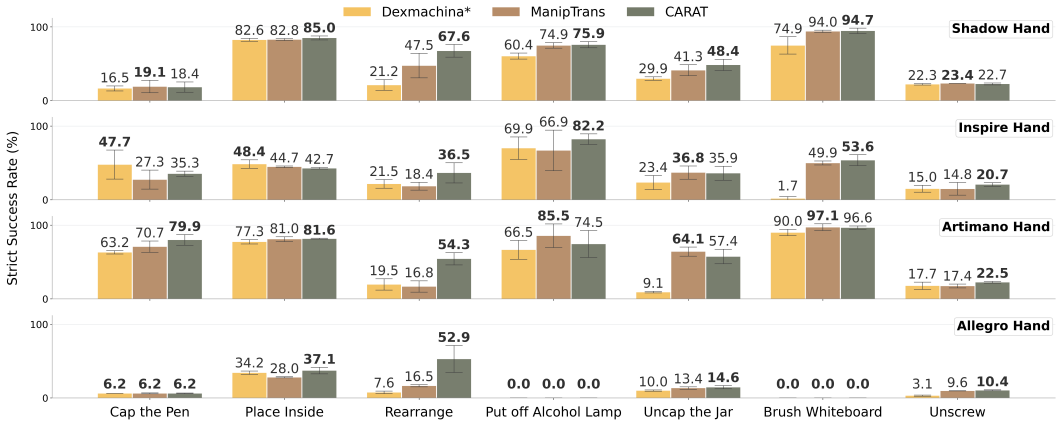


Figure 7: **Pair-wise strict performance across all 28 task-hand pairs.** Bars report the three-seed mean S-SR$_{E1000}$ with ± 1 standard deviation.



Figure 8: **Qualitative CARAT rollouts on representative task-hand pairs.** Each sequence starts from the indicated reference phase and visualizes successful execution of the remaining manipulation suffix to the reference end.

nation achieves the highest average performance on both metrics, supporting their complementary roles in interaction calibration and adaptive reset allocation. Fig. 8 visualizes CARAT rollouts on three representative task-hand pairs. Each rollout starts from an indicated reference phase and executes the remaining trajectory suffix, providing qualitative examples of the learned policy behavior across different manipulation tasks and hand embodiments.

5 Conclusions and Limitations

We presented CARAT, a human-to-robot dexterous manipulation transfer framework that exploits two complementary sources of variation during policy learning. Static Compatibility estimates task-hand-specific prior reusability and calibrates how strongly demonstrated interaction patterns should be preserved. Dynamic Readiness tracks phase-wise competence and reallocates training toward the current learning frontier. Across seven tasks and four heterogeneous dexterous hands, both components improve performance individually, and provide complementary gains when combined. We hope that CARAT offers a useful perspective on exploiting task-, embodiment-, and learner-dependent variation in human-to-robot dexterous manipulation transfer.

CARAT remains limited to simulation and relies on an initial robot reference, predefined semantic correspondences, and hand-crafted compatibility descriptors. Dynamic Readiness uses a coarse four-phase partition and adapts only reset allocation; moreover, our strict protocol evaluates continuation from sampled reference phases, so success does not imply reliable execution from task onset for every task-hand pair. Future work will explore learned compatibility representations, finer-grained readiness and long-horizon progression, broader task and embodiment coverage, and real-robot transfer under perception, dynamics, tactile, and control uncertainties.

Reproducibility Statement

Sections 3.2-3.5 describe the complete CARAT formulation. Appendices A and B provide detailed definitions of Static Compatibility, Soft Interaction Constraint, Dynamic Readiness, and Reset-Mixture Scheduling. Appendix C documents the policy architecture, optimization and simulation settings, baseline implementation, and strict evaluation protocol, while Appendix D describes hyperparameter calibration. Complete task-wise results, learning curves, and scheduler visualizations are provided in Appendices E-G. The source code will be released upon paper acceptance.

AI Use Statement

In this work, we only used generative AI tools to assist with code development and language polishing of the manuscript. All AI-assisted code was reviewed and tested by two authors, and all AI-assisted text was manually checked and revised for accuracy and consistency. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and experimental results.

References

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2729–2749. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/cheng25b.html.

Yunhai Feng, Natalie Leung, Jiaxuan Wang, Lujie Yang, Haozhi Qi, and Preston Culbertson. A minimalist retargeting-guided reinforcement learning recipe for dexterous manipulation. *arXiv preprint arXiv:2607.11874*, 2026.

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9164–9170. IEEE, 2020.

Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2372–2395. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/iyer25a.html.

Kai Junge and Josie Hughes. Adapt-teleop: Robotic hand with human matched embodiment enables dexterous teleoperated manipulation. *npj Robotics*, 3(1):31, 2025.

Yuanchuan Lai, Qing Gao, Ziyan Liang, Xianfeng Cheng, Junjie Hu, and Zhaojie Ju. Dextele: A dual-arm dexterous teleoperation system based on motion retargeting and adaptive force control. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026a.

Yuanchuan Lai, Qing Gao, Ziyan Liang, Junjie Hu, and Zhaojie Ju. Objretarget: An object-aware motion retargeting framework with anthropomorphic arm constraints and polyhedral hand modeling. *arXiv preprint arXiv:2607.03828*, 2026b.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003. IEEE, 2025.

Sizhe Li, Zhiao Huang, Tao Chen, Tao Du, Hao Su, Joshua B. Tenenbaum, and Chuang Gan. Dexdeform: Dexterous deformable object manipulation with human demonstrations and differentiable physics. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=LIV7-_7pYPl.

Yutong Liang, Quanquan Peng, Ri-Zhao Qiu, and Xiaolong Wang. Contrack: Constrained hand motion tracking with adaptive trade-off control. In *European Conference on Computer Vision*, 2026.

Xiaoyi Lin, Kunpeng Yao, Lixin Xu, Xueqiang Wang, Xuetao Li, Yuchen Wang, and Miao Li. Dexflow: A unified approach for dexterous hand pose retargeting and interaction. *arXiv preprint arXiv:2505.01083*, 2025.

Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Parameterized quasi-physical simulators for dexterous manipulations transfer. In *European Conference on Computer Vision*, pp. 164–182. Springer, 2024.

Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *International Conference on Learning Representations*, volume 2025, pp. 85970–85994, 2025.

Haozhe Lou, Mingtong Zhang, Haoran Geng, Hanyang Zhou, Sicheng He, Zhiyuan Gao, Siheng Zhao, Jiageng Mao, Pieter Abbeel, Jitendra Malik, et al. D-rex: Differentiable real-to-sim-to-real engine for learning dexterous grasping. In *International Conference on Learning Representations*, 2026.

Tyler Ga Wei Lum, Olivia Y. Lee, Karen Liu, and Jeannette Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4418–4441. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/lum25a.html.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos. *arXiv preprint arXiv:2606.19333*, 2026.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. Spider: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pp. 570–587. Springer, 2022.

Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 594–605. PMLR, 14–18 Dec 2023a. URL https://proceedings.mlr.press/v205/qin23a.html.

Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023b. doi: 10.15607/RSS.2023.XIX.015.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.043.

Jielin Wu, Shenzhe Yao, Guanqi He, Xiaohan Liu, Zhaoqing Zeng, Xiangrui Jiang, Han Yang, Wentao Zhang, and Hang Zhao. Toporetarget: Interaction-preserving retargeting for dexterous manipulation. *arXiv preprint arXiv:2606.16272*, 2026.

Sirui Xu, Yu-Wei Chao, Liuyu Bian, Arsalan Mousavian, Yu-Xiong Wang, Liangyan Gui, and Wei 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*, pp. 2184–2199. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/xu25d.html.

Yucheng Xu, Xiaofeng Mao, Elle Miller, Xinyu Yi, Yang Li, Zhibin Li, and Robert B. Fisher. Demobot: Efficient learning of bimanual manipulation with dexterous hands from third-person human videos. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

Zhao-Heng Yin, Changhao Wang, Luis Pineda, Krishna Bodduluri, Tingfan Wu, Pieter Abbeel, and Mustafa Mukadam. Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 17376–17382. IEEE, 2025.

Haoqi Yuan, Bohan Zhou, Yuhui Fu, and Zongqing Lu. Cross-embodiment dexterous grasping with reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 81413–81434, 2025.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456. IEEE, 2024.

Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. In *European Conference on Computer Vision*, pp. 386–403. Springer, 2024.

Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 5106–5133. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/zhang25j.html.

Mandi Zhao, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/63277.

Bohan Zhou, Haoqi Yuan, Yuhui Fu, and Zongqing Lu. Learning diverse bimanual dexterous manipulation skills from human demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28919–28927, 2026.

Xinghao Zhu, Zixi Liu, Shalin Jain, Chenran Li, Milad Noori, Michael Andres Lin, Huihua Zhao, John Welsh, Mrinal Verghese, Wei Liu, et al. Learning dexterous manipulation using contact wrench guidance from human demonstration. *arXiv preprint arXiv:2607.00033*, 2026.

A Details of Static Compatibility and Soft Interaction Constraint

A.1 Absolute Hand Scale

We use the thumb and middle-finger wrist-to-tip chains as two complementary scale anchors. The thumb chain captures the scale of the structure supporting opposition, while the middle-finger chain provides a longitudinal hand-size reference. For $f \in \{\text{thumb}, \text{middle}\}$, let ℓ_f^h and ℓ_f^r denote the corresponding human and robot chain lengths. Define $r_f = \frac{\ell_f^r}{\ell_f^h}, \Delta_f = |\log r_f|$, and map the scale mismatch to a bounded discrepancy:

$$g(r_f) = \frac{\Delta_f^4}{\Delta_f^4 + \tau_{\text{size}}^4}, \qquad d_{\text{size}} = \frac{1}{2}\left[g(r_{\text{thumb}}) + g(r_{\text{middle}})\right], \tag{9}$$

where $\tau_{\text{size}} = \log 1.3$. The log-ratio treats reciprocal scale deviations symmetrically, while the bounded mapping suppresses small mismatches and saturates for large ones. The resulting d_{size} summarizes the overall absolute-scale discrepancy between the human and robot hands, with smaller values indicating closer physical scale.

A.2 Relative Skeletal Geometry

We use geodesic distances, measured along the kinematic tree, to compare the relative arrangement of semantic landmarks. Let $L_h(x, x')$ and $L_r(y, y')$ denote the corresponding **normalized** geodesic distances on the human skeleton and robot kinematic tree, respectively. Each distance matrix is normalized by its largest finite entry to remove overall scale. The structural discrepancy is then computed as:

$$d_L = \sum_{x,y,x',y'} \Pi_{xy}\Pi_{x'y'} \left(L_h(x, x') - L_r(y, y')\right)^2, \tag{10}$$

where Π follows the fixed human-robot semantic correspondence, so that only matched landmark pairs contribute to the comparison. Thus, d_L measures the scale-invariant skeletal-layout discrepancy, with lower values indicating more similar relative geometry.

A.3 Kinematic-Path Mobility for Human/Robot Hands

A.3.1 Human Hand Path Mobility

OakInkV2 (Zhan et al., 2024) provides the human hand through a mesh-based parametric representation. For path-mobility computation, each hand is converted to a canonical 21-keypoint skeleton following the MANO ordering. Since the human hand does not have a URDF-style actuation model directly comparable to that of a robot hand, its mobility is characterized using anatomical joint DoFs defined on this skeleton. We therefore use an anatomy-based mobility proxy. For a pair of semantic human nodes $x, x' \in X$, let $\mathcal{P}_h(x, x')$ denote the unique shortest path between them on the canonical hand skeleton. The human path mobility is defined as:

$$C_h(x, x') = \sum_{j \in \mathcal{P}_h(x,x')} d_j, \tag{11}$$

where d_j denotes the anatomical degrees of freedom (DoF) assigned to joint j. Global wrist motion is excluded because the descriptor is intended to capture internal hand mobility rather than the global pose of the hand.

For the four non-thumb fingers, each metacarpophalangeal (MCP) joint is assigned 2 DoFs, corresponding to flexion-extension and abduction-adduction, while each PIP (Proximal interphalangeal joint) and DIP (Distal interphalangeal joint) contributes 1 DoF. For the thumb, the CMC (Carpometacarpal joints), MCP, and IP (Interphalangeal joint of thumb) joints contribute 2, 1, and 1 DoFs, respectively. Fingertips themselves contribute no additional DoF. Thus,

$$d_j = \begin{cases} 2, & j \in \{\text{finger MCP}, \text{thumb CMC}\}, \\ 1, & j \in \{\text{finger PIP/DIP}, \text{thumb MCP/IP}\}, \\ 0, & j \in \{\text{wrist}, \text{fingertip}\}. \end{cases} \tag{12}$$

As a concrete example, consider the path from the wrist to the index fingertip. The path passes through the index MCP, PIP, and DIP joints before reaching the fingertip. Hence,

$C_h(\text{wrist, index tip}) = 2 + 1 + 1 + 0 = 4$. Similarly, the index-tip-to-thumb-tip path gives $C_h(\text{index tip, thumb tip}) = 0 + 1 + 1 + 2 + 2 + 1 + 1 + 0 = 8$. These assignments provide a compact anatomical approximation of the motion capability available along each human-hand path.

A.3.2 Robot Hand Path Mobility

For a robot hand, directly counting URDF joints or motors is insufficient. Considering the worst situation, a hand may contain many joints while some are mechanically coupled, underactuated, or produce highly correlated motions. Consequently, the number of joints does not necessarily reflect the number of motions that can be independently generated along a kinematic path. For a pair of robot nodes $y, y' \in Y$, let $\mathcal{P}_r(y, y')$ denote the corresponding path in the robot kinematic tree. Suppose this path contains N joints but only M independent actuators. Their relationship is represented by the actuation matrix $A_{yy'} \in \mathbb{R}^{N\times M}$, such that $\dot{q}_{\text{path}} = A_{yy'}\dot{u}$, where $\dot{u} \in \mathbb{R}^M$ contains independent actuator velocities and $\dot{q}_{\text{path}} \in \mathbb{R}^N$ contains the resulting joint velocities. Directly actuated joints correspond to independent columns of $A_{yy'}$, whereas mimic or coupled joints are represented through the corresponding linear dependencies. This construction allows the same formulation to handle direct-drive, coupled, and underactuated hand designs. The role of this matrix in the original implementation is to distinguish nominal joint count from effective actuation.

Relative Pose Path Mobility. At a valid robot configuration $q^{(k)}$, the relative spatial Jacobian $J_{yy'}^{\text{rel},(k)} \in \mathbb{R}^{6\times N}$ maps joint velocities along the path to the relative translational and rotational velocity between nodes y and y'. Combining the kinematic and actuation mappings gives

$$J_{yy'}^{\text{act},(k)} = J_{yy'}^{\text{rel},(k)} A_{yy'}^{(k)}, \quad \dot{\xi}_{yy'} = J_{yy'}^{\text{act},(k)}\dot{u}, \tag{13}$$

where $\dot{\xi}_{yy'} \in \mathbb{R}^6$ is the relative spatial velocity between the two nodes. Intuitively, each linearly independent column direction of $J_{yy'}^{\text{act},(k)}$ corresponds to one independent way in which the available actuators can change the relative pose of the two nodes. Its effective rank is therefore defined as

$$r_{yy'}^{(k)} = \text{rank}_\epsilon\left(J_{yy'}^{\text{rel},(k)} A_{yy'}^{(k)}\right), \tag{14}$$

where rank_ϵ is computed using singular-value decomposition (SVD): $\text{rank}_\epsilon(J) = \sum_i \mathbb{I}[\sigma_i(J) > \epsilon], \epsilon = 10^{-3}$. Since the relative motion contains three translational and three rotational dimensions, $0 \le r_{yy'}^{(k)} \le 6$. Thus, $r_{yy'}^{(k)}$ measures how many independent ways the robot can move node y relative to y' using the available actuators, up to the six dimensions of relative translation and rotation.

Internal Motion Path Mobility. The relative Jacobian only describes how the two endpoint nodes can move with respect to each other. Once this relative motion reaches full 6D rank, it cannot reveal whether additional motion remains within the intermediate links. For example, two kinematic paths may both achieve full relative pose control, while one still allows internal links to change configuration without losing this endpoint capability. To capture this additional mobility, the Jacobians of all links along $\mathcal{P}_r(y, y')$ are stacked:

$$J_{\text{full}}^{(k)} = \begin{bmatrix} J_{\ell_1}^{(k)} \\ \vdots \\ J_{\ell_P}^{(k)} \end{bmatrix}, \qquad J_{\text{full,act}}^{(k)} = J_{\text{full}}^{(k)} A_{yy'}^{(k)}. \tag{15}$$

Each link admits its own geometric Jacobian because its spatial velocity depends differently on the upstream joints. Specifically, J_ℓ maps joint velocities to the 6D spatial velocity of link ℓ. Stacking these Jacobians therefore exposes actuator-induced motion throughout the entire kinematic path, rather than only at the two endpoints. The effective rank of $J_{\text{full,act}}^{(k)}$

$$m_{yy'}^{(k)} = \text{rank}_\epsilon\left(J_{\text{full,act}}^{(k)}\right) \tag{16}$$

measures how many independent actuator-induced motion patterns are expressed across the entire path. In contrast, $r_{yy'}^{(k)}$ observes only the relative motion of the two endpoint nodes. Therefore, when $r_{yy'}^{(k)} = 6$, any additional dimensions captured by $m_{yy'}^{(k)}$ indicate internal mobility that is invisible from the endpoint-relative Jacobian alone.

CARAT combines the endpoint-relative and internal-motion terms as

$$C_r^{(k)}(y, y') = r_{yy'}^{(k)} + \mathbb{I}\left[r_{yy'}^{(k)} = 6\right]\left[m_{yy'}^{(k)} - 6\right]_+. \tag{17}$$

The first term measures the available relative-pose mobility. The second term credits additional internal actuation dimensions only after full 6D relative motion is available. Importantly, this full-rank gate is a design choice of our descriptor rather than a general definition of kinematic redundancy. Its purpose is to prioritize full relative-pose capability before crediting additional internal mobility. This avoids assigning a high score to mechanisms with many actuators when their endpoint motion is still restricted to only a few independent directions.

Multi-Configuration Evaluation. Jacobian rank depends on robot configuration and can decrease at kinematic singularities. Computing the descriptor at a single pose may therefore underestimate the mobility supported by the underlying mechanism. To reduce this sensitivity, K valid configurations are sampled from the robot joint-limit space, while respecting the specified actuation and coupling relationships. The robot path descriptor is then defined by the maximum value observed across these configurations:

$$C_r(y, y') = \max_{k=1,\dots,K} C_r^{(k)}(y, y'), \tag{18}$$

where $K = 10$. We use the maximum rather than the average to estimate the generic structural mobility available to the mechanism. Multi-configuration sampling reduces the probability that an otherwise movable path receives a low score merely because the sampled configuration lies close to a singularity.

A.4 Task-Specific Interaction Discrepancy

Task-Specific Interaction Discrepancy measures how closely the robot reference preserves the demonstrated hand-object interaction over the aligned trajectory. The comparison is performed over fixed semantic human-robot landmark correspondences and uses both hand configuration and local contact geometry.

Frame-Level Interaction Quantities. For each matched human-robot landmark pair (x, y) at frame t, four quantities are considered: $p_{t,x}^h, c_{t,x}^h, n_{t,x}^h, v_{t,x}^h$ for the human hand, and $p_{t,y}^r, c_{t,y}^r, n_{t,y}^r, v_{t,y}^r$ for the robot hand.

- $p \in \mathbb{R}^3$ is the 3D position of the semantic hand landmark, such as the wrist or a fingertip;
- $c \in \mathbb{R}^3$ is the object-surface contact point associated with that landmark;
- $n \in \mathbb{R}^3$ is the unit surface normal at the associated contact point;
- $v \in \{0, 1\}$ indicates whether a valid contact is present at the current frame.

Contact-Validity Gating. Contact location and surface normal are meaningful only when both matched landmarks have valid contact. A joint contact-validity indicator is therefore defined as $b_{t,xy} = v_{t,x}^h v_{t,y}^r$. Thus, $b_{t,xy} = 1$ only when both the human and robot landmarks are in valid contact at frame t.

Per-Frame Discrepancy. For each aligned frame, the interaction discrepancy contains four complementary terms:

$$\begin{aligned}
E_{\text{pose},t} &= \sum_{x,y} \Pi_{xy} \left(\frac{\left\| p_{t,x}^h - p_{t,y}^r \right\|_2}{\sigma_{\text{pose}}} \right)^2, \\
E_{\text{contact},t} &= \sum_{x,y} \Pi_{xy} b_{t,xy} \left(\frac{\left\| c_{t,x}^h - c_{t,y}^r \right\|_2}{\sigma_{\text{contact}}} \right)^2, \\
E_{\text{normal},t} &= \sum_{x,y} \Pi_{xy} b_{t,xy} \left(1 - \left\langle n_{t,x}^h, n_{t,y}^r \right\rangle \right), \\
E_{\text{mask},t} &= \sum_{x,y} \Pi_{xy} \left| v_{t,x}^h - v_{t,y}^r \right|^2.
\end{aligned} \tag{19}$$

Here, Π_{xy} follows the fixed semantic human-robot correspondence introduced in the main text (Sec. 3.3). The meaning of each term is described as follows:

- $E_{\mathrm{pose},t}$ measures the Euclidean mismatch between corresponding semantic hand landmarks. Because it is evaluated at every aligned frame, it reflects the time-varying mismatch of the overall hand configuration. The normalization factor σ_{pose} converts metric position error into a dimensionless quantity.
- $E_{\mathrm{contact},t}$ compares where the human and robot interact with the object surface. It is evaluated only when both corresponding landmarks are in valid contact. The scale $\sigma_{\mathrm{contact}}$ plays the same normalization role for contact-point error.
- $E_{\mathrm{normal},t}$ compares the local surface orientations at the matched contact locations. Since both normals are unit vectors, $1 - \langle n^h_{t,x}, n^r_{t,y} \rangle$ is zero when the two normals coincide and increases as their directions diverge. This term therefore captures a part of the local contact geometry.
- $E_{\mathrm{mask},t}$ explicitly penalizes disagreement in contact existence. It becomes nonzero when a corresponding landmark is in contact on one side but not the other, thereby capturing missing or additional contacts.

The implementation uses $\sigma_{\mathrm{pose}} = 0.04\,\mathrm{m}, \sigma_{\mathrm{contact}} = 0.01\,\mathrm{m}$. These values are fixed across all task-hand pairs.

Trajectory Aggregation and Normalization. The frame-level discrepancies are averaged over the set of valid aligned frames $\mathcal{V}$ to obtain the trajectory-level interaction error

$$E_{\mathrm{int}} = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \left(E_{\mathrm{pose},t} + E_{\mathrm{contact},t} + E_{\mathrm{normal},t} + E_{\mathrm{mask},t} \right). \tag{20}$$

Because E_{int} is unbounded, we compress it into the normalized Task-Specific Interaction Discrepancy

$$D_{\mathrm{I}} = \frac{\tanh(\mathrm{clip}(E_{\mathrm{int}}, 0, E_{\max})/\kappa)}{\tanh(E_{\max}/\kappa)}, \qquad D_{\mathrm{I}} \in [0, 1], \tag{21}$$

where $E_{\max} = 20$ and $\kappa = 4$ for all task-hand pairs. The clipping suppresses the influence of extreme interaction errors, while the $\tanh$ mapping provides a smooth bounded compression. Consequently, $D_{\mathrm{I}} = 0$ indicates perfect preservation of the demonstrated interaction under the defined terms, whereas larger D_{I} indicates greater task-specific interaction mismatch.

A.5 Soft Interaction Constraint

Compatibility-Calibrated Tolerance. Static Compatibility determines how far the robot is allowed to deviate from the demonstrated local fingertip-object interaction. For each active hand side $s \in \mathrm{S}$, the corresponding static discrepancy $D_{\mathrm{S}}^{(s)} \in [0, 1]$ is mapped to a continuous interaction tolerance radius $\rho_{cont}^{(s)}$:

$$\rho_{\mathrm{cont}}^{(s)} = 10 + 30 D_{\mathrm{S}}^{(s)} \quad (\mathrm{mm}), \tag{22}$$

yielding a tolerance range of 10 mm - 40 mm. A smaller $D_{\mathrm{S}}^{(s)}$, indicating higher compatibility, produces a tighter tolerance region; a larger discrepancy produces a wider region and allows greater deviation from the demonstrated interaction geometry. For a task-hand pair, the side-wise radii are averaged and quantized to $1\,\mathrm{mm}$:

$$\rho_{\mathrm{pair}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \rho_{\mathrm{cont}}^{(s)}, \qquad \rho = (1\mathrm{mm})\,\mathrm{RoundHalfUp}\left(\frac{\rho_{\mathrm{pair}}}{1\mathrm{mm}} \right). \tag{23}$$

For single-hand tasks, this reduces to the radius of the active side; for bimanual tasks, the left- and right-hand radii are averaged. The resulting ρ is computed before policy learning and shared by all active sides and valid semantic fingertips throughout training. Hence, the soft interaction constraint tolerance itself remains fixed as the policy evolves.

Object-Centric Interaction Error. For a fixed active side s, we omit the side index for clarity. Let $j^h_{t,i}$ denote the i-th demonstrated human fingertip and $y^e_{t,i}$ its semantically corresponding robot

fingertip at the current rollout state. Let $n(i)$ denote the object part associated with fingertip i, with reference pose $(p_{t,n(i)}, R_{t,n(i)}) \in g_t$ and current rollout pose $(\hat{p}_{t,n(i)}, \hat{R}_{t,n(i)}) \in \hat{g}_t$. The demonstrated fingertip is first expressed relative to the reference object, and then transported through the current object pose:

$$\bar{j}_{t,i} = R_{t,n(i)}^{\top}\big(j_{t,i}^{h} - p_{t,n(i)}\big), \qquad \tilde{j}_{t,i} = \hat{p}_{t,n(i)} + \hat{R}_{t,n(i)}\bar{j}_{t,i}. \tag{24}$$

We then define $e_{t,i} = \left\| y_{t,i}^{e} - \tilde{j}_{t,i} \right\|_2$ to compare the robot fingertip location relative to the current object pose with the demonstrated human fingertip location relative to the reference object pose. This object-centric construction prevents global object translation or rotation from being repeatedly counted as fingertip interaction error.

Interaction-Phase Gating. The additional constraint is activated only when the reference trajectory enters a meaningful interaction phase. Let $d_{t,i}^{\mathrm{ref}}$ denote the reference distance between fingertip i and its associated object part. We define

$$a_{t,i} = \mathrm{clip}\left(\frac{d_{\mathrm{off}} - d_{t,i}^{\mathrm{ref}}}{d_{\mathrm{off}} - d_{\mathrm{on}}}, 0, 1\right), \qquad d_{\mathrm{on}} = 0.02\,\mathrm{m}, \quad d_{\mathrm{off}} = 0.05\,\mathrm{m}. \tag{25}$$

Thus, the constraint is fully active within $2\,\mathrm{cm}$, inactive beyond $5\,\mathrm{cm}$, and linearly interpolated in between. This gate only controls the proposed Soft Interaction Constraint (SIC) term. When multiple semantic fingertips correspond to the same robot body, directly summing their contributions would over-weight the shared link. Let $m_{body(i)}$ denote the number of semantic landmarks mapped to the robot body $body(i)$. We therefore use the corrected weight $\tilde{a}_{t,i} = \frac{a_{t,i}}{m_{body(i)}}$.

Tolerance Violation and Reward Composition. SIC penalizes only deviations beyond the compatibility-calibrated radius. We first define the normalized excess $z_{t,i} = \left[\frac{e_{t,i}}{\rho} - 1\right]_+$, and apply a pseudo-Huber penalty to realize side-wise SIC:

$$\phi_\delta(z) = \delta^2\left(\sqrt{1 + (z/\delta)^2} - 1\right), \qquad \mathcal{L}_{\mathrm{SIC},t}^{(s)} = \frac{\sum_i \tilde{a}_{t,i}\phi_\delta(z_{t,i})}{\max\left(\sum_i \tilde{a}_{t,i}, \varepsilon\right)}. \tag{26}$$

We use $\delta = 0.25, \lambda = 20, \varepsilon = 10^{-6}$ for all task-hand pairs, without reward warm-up. When all active fingertips lie within the tolerance region, $z_{t,i} = 0$ and no additional penalty is incurred. Once the boundary is exceeded, the penalty grows smoothly with the relative tolerance violation $e_{t,i}/\rho - 1$, rather than with an absolute metric distance. The SIC term is added after computing the original per-side ManipTrans (Li et al., 2025) reward:

$$\mathcal{R}_{\mathrm{CARAT},t} = \sum_{s\in\mathcal{S}}\left(\mathcal{R}_{\mathrm{base},t}^{(s)} - \lambda\mathcal{L}_{\mathrm{SIC},t}^{(s)}\right) = \mathcal{R}_{\mathrm{base},t} - \lambda\sum_{s\in\mathcal{S}}\mathcal{L}_{\mathrm{SIC},t}^{(s)}. \tag{27}$$

Thus, SIC augments the baseline with a non-positive, compatibility-calibrated interaction-geometry penalty without removing or reweighting any original ManipTrans reward term.

B Details of Dynamic Readiness and Reset-Mixture Scheduling

B.1 Fixed Tracking Schedule and Look-Ahead Evaluation

Throughout Dynamic Readiness and Reset-Mixture Scheduling (DR-RMS) training, gravity and friction are fixed at their target simulation values. The tracking curriculum follows the original fixed ManipTrans (Li et al., 2025) schedule. Let f denote the elapsed simulator frames and $q \in [0.7, 1.0]$ the tracking-boundary scale, where a larger q corresponds to more relaxed tracking tolerances. The schedule is

$$q_{\mathrm{fixed}}(f) = 0.7 + 0.3(2e)^{-f/3200}. \tag{28}$$

Thus, $q_{\mathrm{fixed}}(0) = 1.0$ and progressively approaches the final strict boundary $q = 0.7$. Object-position and object-rotation thresholds scale cubically with q,

$$\tau_{\mathrm{obj,pos}}(q) = 0.02\left(\frac{q}{0.7}\right)^3, \qquad \tau_{\mathrm{obj,rot}}(q) = 30^\circ\left(\frac{q}{0.7}\right)^3, \tag{29}$$

whereas hand-tracking thresholds scale linearly with q.

Dynamic Readiness is updated every $\Delta f = 320$ simulator frames. At decision frame f_k, we probe the tracking boundary that the fixed schedule will use at the next decision point:

$$q_k^+ = \max\left(0.7,\, q_{\text{fixed}}(f_k + \Delta f)\right). \tag{30}$$

This look-ahead boundary is used only for competence estimation and never modifies or delays the fixed tracking schedule. For every completed episode n, we record its actual success label $\xi_n \in \{0, 1\}$ and a critical boundary value $q_{\text{crit}}^{(n)}$. At each simulation step, q_{required} denotes the smallest boundary scale under which all recorded hand- and object-tracking errors would remain valid; the episode-level critical value is $q_{\text{crit}}^{(n)} = \max_t q_{\text{required},t}^{(n)}$. A trajectory requiring a larger q_{crit} therefore depends on a more relaxed tracking boundary. We define the binary look-ahead success label

$$\xi_{n,k}^+ = \xi_n \, \mathbb{I}\left[q_{\text{crit}}^{(n)} \le q_k^+\right]. \tag{31}$$

Hence, $\xi_{n,k}^+ = 1$ only when the episode succeeds under its actual training conditions and its recorded trajectory would remain valid under the tighter boundary scheduled for the next decision interval. This counterfactual check reuses recorded tracking errors and requires no additional rollout.

B.2 Dynamic Readiness Estimation: Phase-Wise Candidate Competence

We partition each reference trajectory into $B = 4$ equal intervals according to its normalized starting time. For episode n, let f_n denote its zero-indexed starting reference frame and L_n the reference length. Its phase is:

$$b_n = 1 + \min\left(\left\lfloor B \frac{f_n}{\max(L_n - 1, 1)} \right\rfloor, B - 1\right), \qquad B = 4. \tag{32}$$

These bins are temporal partitions rather than manually specified semantic stages. Competence for phase b measures the ability to complete the remaining reference trajectory when initialized from that phase.

Within each 320-frame decision window, only episodes that have actually terminated are included in the statistics. For phase b, let $N_b = \sum_n \mathbb{I}[b_n = b]$ be the number of completed episodes, and $M_b = \sum_n \mathbb{I}[b_n = b]\, \xi_{n,k}^+$ the number that both succeed and satisfy the next-interval tracking boundary. The phase-wise candidate competence is $\widehat{C}_b = \frac{M_b}{N_b}$. Thus, $\widehat{C}_b$ jointly reflects whether the policy can complete the remaining task and whether its successful behavior is sufficiently robust to the immediately upcoming tracking requirement. To reduce sensitivity to finite-sample fluctuations, scheduling uses a one-sided Wilson lower confidence bound. We define $\underline{C}_b = \mathrm{LCB}(M_b, N_b; z_w = 1.645)$:

$$\underline{C}_{b,k} = \frac{\widehat{C}_{b,k} + \frac{z_{\text{W}}^2}{2N_{b,k}} - z_{\text{W}}\sqrt{\frac{\widehat{C}_{b,k}(1-\widehat{C}_{b,k})}{N_{b,k}} + \frac{z_{\text{W}}^2}{4N_{b,k}^2}}}{1 + \frac{z_{\text{W}}^2}{N_{b,k}}}, \tag{33}$$

where $z_{\text{W}} = 1.645$ gives a 95% one-sided Wilson lower confidence bound. If $N_{b,k} = 0$, we set $\underline{C}_{b,k} = 0$. The phase-wise Dynamic Readiness is defined as:

$$A_b = \mathrm{clip}\left(\frac{\underline{C}_b}{c_{\text{front}}}, 0, 1\right), \qquad c_{\text{front}} = 0.20. \tag{34}$$

Therefore, $A_b = 1$ indicates that the conservative candidate competence of phase b has reached the minimum level required for frontier identification; it does not imply full task mastery.

B.3 Readiness-Guided Reset-Mixture Scheduling

Competence Gate and Learning-Frontier Identification. Before introducing any frontier-biased resets, we require sufficient phase-wise evidence. Each of the four phases must contain at least

$N_{\min} = 64$ completed episodes in the current decision window. We then compute the balanced and worst-phase candidate competence:

$$C_{\text{bal}} = \frac{1}{4}\sum_{b=1}^{4} \underline{C}_b, \qquad C_{\min} = \min_{b\in\{1,\dots,4\}} \underline{C}_b. \tag{35}$$

The competence gate is considered satisfied when: $N_b \geq 64 \quad \forall b, C_{\text{bal}} \geq 0.50, C_{\min} \geq 0.20$. The average criterion ensures sufficient overall competence, while the minimum criterion prevents strong later phases from masking a severely under-trained earlier phase. Once this gate is satisfied, no additional frontier sampling is applied and training returns entirely to the baseline reset mechanism. If phase coverage is sufficient but the competence gate is not yet satisfied, we search for the current learning frontier. Phase $b \in \{1, 2, 3\}$ is a valid frontier candidate when $\underline{C}_b < c_{\text{front}} \leq \underline{C}_{b+1}$, with sufficient evidence on both sides. The adjacent readiness gap is $G_b = [A_{b+1} - A_b]_+$. Among valid candidates, we select $b^* \in \arg\max_b G_b$. The selected phase is therefore the weaker phase immediately preceding a successor that has already reached basic competence. If several candidates have identical gaps, the later phase in the reference trajectory is selected. If no valid candidate exists, no frontier injection is performed. This criterion deliberately avoids selecting the globally weakest phase: a phase that is far outside the learner's current capability may provide little useful training signal. Instead, b^* approximates the local boundary between currently attainable and not-yet-reliable trajectory regions.

Reset-Mixture Scheduling. Let $p_{\text{base}}(b)$ denote the phase-level distribution induced by the original ManipTrans reference state initialization (RSI) sampler. When phase coverage is sufficient, the competence gate remains unmet, and a valid frontier b^* is identified, we use

$$p_{\text{train}}(b) = (1-\beta)p_{\text{base}}(b) + \beta\mathbb{I}[b = b^*], \qquad \beta = 0.20. \tag{36}$$

Otherwise, $p_{\text{train}}(b) = p_{\text{base}}(b)$. In implementation, the $1-\beta$ branch invokes the original ManipTrans reset sampler unchanged, whereas the β branch samples uniformly from valid reference starting frames within the selected frontier phase b^*. Thus, DR-RMS preserves the original reset behavior while injecting additional starts only at the identified learning frontier. The scheduler affects only subsequent episode resets and never interrupts ongoing rollouts. Each parallel environment resamples independently at its next reset and is not permanently assigned to a particular phase.

Separation from the Tracking Curriculum. Dynamic Readiness controls where training starts, not how quickly the task difficulty changes. Throughout training, $q(f) = q_{\text{fixed}}(f)$ is determined solely by simulator time. DR-RMS cannot pause, accelerate, or reverse the tracking schedule, and gravity and friction remain at their target values from the start of training.

C Experimental Setup Details

C.1 Details of Training

Policy Architecture. CARAT reuses the same pretrained and frozen ManipTrans (Li et al., 2025) trajectory imitator introduced in Sec. 3.3, but its role during policy learning differs from the object-aligned replay used to construct τ_r^0. For offline Static Compatibility estimation, τ_r^0 is obtained by replaying the frozen imitator while clamping the object pose to the time-aligned reference. During policy learning, the object instead evolves freely under simulation dynamics, and the same frozen imitator serves only as the base-action generator. PPO optimizes the task-specific residual actor-critic, while the imitator parameters remain fixed; CARAT introduces no additional policy inputs or network branches. During training, base actions are sampled from the frozen imitator distribution and combined with the residual policy using the original ManipTrans action-composition rule:

$$\tilde{a}^{\text{joint}} = \text{clip}\left(a^{\text{base}} + 2a^{\text{res}}, -1, 1\right), \tag{37}$$

with the same action-smoothing and low-level control pipeline. The observation encoder follows ManipTrans's SimpleFeatureFusion, using three 512-dimensional hidden layers followed by a 256-dimensional output with SiLU activations. The actor and critic then share an MLP with widths $[512, 128, 64]$; separate linear heads predict the residual-action mean and scalar value.

Optimization and Training Budget. We use the ManipTrans PPO implementation. The key hyperparameters are summarized in Table 4. Each task-hand pair is trained for 1000 epochs with 4096

parallel environments and three training seeds $\{42, 43, 44\}$. Each PPO epoch collects 32 steps per environment. All formal CARAT runs use a fixed training budget with no performance-based early stopping. The learning rate uses the same ManipTrans warm-up/adaptive schedule: a 10-epoch warm-up is followed by KL-dependent adjustment around the target 0.008.

Table 4: **PPO training hyperparameters.**

Hyperparameter	Value
Parallel environments	4096
Training epochs	1000
Rollout horizon	32
Optimizer	Adam
Learning rate	2×10^{-4}
Discount γ	0.99
GAE λ	0.95
PPO clip ratio	0.2
Critic loss coefficient	4.0
Entropy coefficient	0.0
Gradient-norm clip	1.0
Mini-batch size	1024
Optimization passes / epoch	5
KL target	0.008

Simulation and Fixed Components. All experiments use GPU PhysX in Isaac Gym with a $1/60\,\mathrm{s}$ control timestep and two physics substeps per control step. We use a RTX 4090D GPU for all experiments. CARAT keeps the ManipTrans robot assets, control interface, low-level action execution, network architecture, and PPO optimizer unchanged. The frozen imitator and its normalization statistics are not updated during policy learning. Static Compatibility and the resulting SIC tolerance are computed before training and remain fixed, whereas Dynamic Readiness updates only the reset/start-frame distribution online. For CARAT, gravity and object friction use their target values from the beginning of training.

C.2 Implementation Details of DexMachina

Definition. DexMachina* denotes our adaptation of the core Virtual Object Controller (VOC) curriculum of DexMachina (Zhao et al., 2026) to the ManipTrans backbone. It is not a direct execution of the original DexMachina codebase. Instead, we retain the ManipTrans observations, frozen trajectory imitator, residual-PPO policy, action interface, base reward, robot/object assets, and task termination criteria, and introduce only the VOC-based object assistance and its curriculum. The superscript * distinguishes this controlled adaptation from the original DexMachina system.

Virtual Object Controller. DexMachina* preserves the core VOC principle of the original method: during early training, the manipulated object receives an auxiliary wrench that drives it toward the time-aligned reference trajectory, and this assistance is progressively removed as policy competence improves. Because the original DexMachina implementation relies on Genesis-specific gain control, we implement the equivalent mechanism in Isaac Gym as a world-frame force/torque controller based on object pose and velocity errors. We denote the VOC assistance strength by $\eta \in [0, 1]$, where $\eta = 1$ gives maximum assistance and $\eta = 0$ disables the auxiliary wrench entirely. Translational and rotational feedback use critically damped second-order controllers with nominal frequencies $f_p = f_R = 3\,\mathrm{Hz}$ and damping ratios $\zeta_p = \zeta_R = 1$. As η decreases, the controller stiffness and wrench limits are reduced accordingly; the maximum linear and angular acceleration limits are $30\,\mathrm{m/s^2}$ and $100\,\mathrm{rad/s^2}$, respectively. At $\eta = 0$, the additional force and torque are explicitly set to zero, so the object evolves under unassisted simulation dynamics.

VOC Curriculum. The assistance level follows the discrete path $\eta : 1.0 \to 0.9 \to \cdots \to 0.1 \to 0.05 \to 0.0$. Transitions occur only between adjacent levels. For each proposed reduction in η, the controller proceeds through candidate adaptation and verification; successful candidates are accepted, whereas failed candidates recover to the previously confirmed assistance level. Curriculum transitions occur only at PPO-update boundaries, avoiding changes to the physical assistance within an ongoing trajectory. The competence thresholds used by the VOC scheduler are calibrated be-

fore formal multi-seed training and then frozen across seeds. The main scheduling parameters are summarized in Table 5.

Table 5: **DexMachina* VOC curriculum parameters.**

Parameter	Value
Episode window size	2048
Min. updates / confirmed stage	100
Candidate adaptation updates	40
Candidate natural episodes	2048
Candidate verification blocks	2
Recovery updates	40
Max. candidate attempts / stage	3
Min. zero-assistance updates	100
Hard update cap	3820

Controlled Backbone and Reward. To isolate the contribution of the VOC curriculum, DexMachina* uses the unchanged ManipTrans imitation reward. The frozen imitator remains the base-action generator and the learned controller remains the ManipTrans residual PPO policy. The failure boundary is fixed at the final strict scale 0.7 throughout DexMachina* training and is explicitly decoupled from the VOC assistance level: $q_{\text{train}} = q_{\text{eval}} = 0.7$. Thus, η changes only the auxiliary object wrench and never relaxes or tightens the tracking-success criterion.

C.3 Strict Evaluation Protocol

Deterministic Evaluation Setup. We evaluate frozen policy checkpoints using a deterministic rollout protocol designed to measure whether the learned policy can stably reproduce the remaining hand-object interaction from different points along the reference trajectory. Exploration is disabled during evaluation: both the frozen imitator and residual policy use their mean actions. All training curricula and physical randomization are disabled, and evaluation uses the target physical parameters and fixed final tracking thresholds. Each evaluation episode is initialized from a pre-specified reference frame. For our unified evaluation protocol, each checkpoint is evaluated on a fixed bank of 256 start frames, and the same bank is reused across checkpoints and methods. Each parallel environment contributes one episode outcome.

Strict Tracking Failure Criteria. To ensure that performance differences are not introduced by a modified evaluation criterion, we retain the original ManipTrans (Li et al., 2025) strict imitation failure gate unchanged, with the strict scale fixed to 0.7. This criterion requires the policy to remain within prescribed hand- and object-tracking tolerances throughout the evaluated trajectory. After the initialization grace period, exceeding any tracking boundary at any timestep immediately marks the episode as failed; errors are neither temporally averaged nor judged solely from the terminal state. At strict scale 0.7, the resulting tracking thresholds are described in Table 6. For the hand, each group error is computed as the mean Euclidean positional error over the corresponding joints or fingertips. Object tracking uses Euclidean position error and the absolute relative rotation angle.

Table 6: **Tracking thresholds (strict scale $= 0.7$).**

Tracking quantity	Maximum allowed error
Object position	$2\,\text{cm}$
Object rotation	30°
Thumb-tip group	$4\,\text{cm}$
Index-tip group	$4.5\,\text{cm}$
Middle-tip group	$5\,\text{cm}$
Ring-tip group	$6\,\text{cm}$
Little-finger-tip group	$6\,\text{cm}$
Level-1 finger-joint group	$7\,\text{cm}$
Level-2 finger-joint group	$8\,\text{cm}$

Contact-Consistency Check. In addition to geometric tracking, the strict evaluator enforces consistency with demonstrated fingertip-object contacts. A reference fingertip is considered contact-active when its reference distance to the object is below $5\,\mathrm{mm}$. For each such fingertip, the evaluator checks whether nonzero simulated contact force has been observed within the most recent three simulation steps. If no contact is detected throughout this window, a contact mismatch is triggered and the episode is marked as failed. The three-step history reduces sensitivity to transient single-step contact fluctuations while still requiring the robot to reproduce the reference contact structure. The contact-history buffer is initialized as active at reset, avoiding spurious failures before the simulator has populated the initial contact-force measurements.

Initialization Grace Period. The first eight simulation steps after each reset form an initialization grace period. During this period, hand-tracking, object-tracking, and contact-consistency violations do not activate the strict failure gate, allowing the simulated hand-object state to settle after initialization.

Episode-Level Strict Success. An episode is counted as successful only when

$$\mathrm{Success} = \mathrm{ReachedReferenceEnd} \ \wedge \ \neg\,\mathrm{StrictFailedEver}.$$

The rollout is allowed to terminate within a margin of three reference frames before the exact trajectory endpoint. Any strict tracking, contact, or numerical failure occurring earlier invalidates the entire episode. For bimanual tasks, both participating sides must satisfy the strict success condition:

$$\mathrm{Success}_{\mathrm{bi}} = \mathrm{Success}_L \wedge \mathrm{Success}_R.$$

Thus, failure of either hand causes the complete bimanual episode to fail.

Checkpoint-Level Strict Success Rate. For a frozen checkpoint, the strict success rate is

$$\mathrm{SR}_{\mathrm{strict}} = \frac{N_{\mathrm{success}}}{N_{\mathrm{eval}}}, \tag{38}$$

where $N_{\mathrm{eval}} = 256$ under the unified fixed-bank protocol. Each evaluation episode starts from one reference start frame in the fixed evaluation bank. For each task-hand pair and method, we train three independent seeds (42,43,44) and report the mean epoch-1000 strict success rate:

$$\overline{\mathrm{SR}}_{E1000} = \frac{1}{3} \sum_{s\in\{42,43,44\}} \mathrm{SR}_{s,E1000}. \tag{39}$$

This seed-averaged quantity is reported as S-SR$_{E1000}$ in the main text. Error bars report the sample standard deviation across the three training seeds, and therefore reflect training variability rather than episode-sampling uncertainty.

Strict AUC$_{1000}$. To evaluate learning efficiency, we independently evaluate checkpoints at $e_k \in \{100, 200, \ldots, 1000\}$. For each training seed, Strict AUC$_{1000}$ is computed as the normalized trapezoidal area under the strict-success curve:

$$\text{S-AUC}_{1000} = \frac{1}{900} \sum_{k=1}^{9} \frac{\mathrm{SR}_{e_k} + \mathrm{SR}_{e_{k+1}}}{2} \left(e_{k+1} - e_k\right). \tag{40}$$

Because the area is normalized by the interval length $1000 - 100 = 900$, S-AUC$_{1000} \in [0, 1]$ and can be interpreted as the average strict-success level over training from epoch 100 to epoch 1000. It captures not only final performance but also learning speed, intermediate performance, and potential late-stage degradation. We report the mean S-AUC$_{1000}$ across the three training seeds.

Protocol Consistency Across Methods. All methods are evaluated using the same deterministic action protocol, strict failure criteria, target physical parameters, and fixed start-frame bank. Therefore, differences in reported strict success rates arise from the learned policies.

D Details of Alpha Calibration

Calibration Subset and Fixed-ρ Lookup. To calibrate the fusion coefficient α in Static Compatibility, we use the 12 task-hand pairs shown in Fig. 5. The subset contains four hands for *Cap the*

Pen, four hands for *Rearrange*, *Allegro* and *Artimano* for *Place Inside*, and *Artimano* and *Inspire* for *Uncap the Jar*. For every pair, we train the ManipTrans baseline and three fixed-tolerance SIC variants with $\rho \in \{10, 25, 40\}$ mm, using the same frozen imitator, PPO backbone, training budget, and seeds $\{42, 43, 44\}$. The only treatment difference is whether SIC is applied and, when enabled, which fixed radius is used. Calibration uses the training-time Online AUC_{1000} (O-AUC_{1000}). Let s_e denote the online success rate logged at epoch e, computed from the most recent completed training episodes. For each seed,

$$\text{O-AUC}_{1000} = \frac{1}{999} \sum_{e=1}^{999} \frac{s_e + s_{e+1}}{2}. \tag{41}$$

The reported value for each task-hand-radius configuration is the mean over the three training seeds.

Selection of Coefficient α. For each active side, Static Compatibility under candidate fusion coefficient α is computed as $D_{\text{S}}^{(s)}(\alpha) = D_{\text{M}}^{(s)} + \alpha(1 - D_{\text{M}}^{(s)})D_{\text{I}}^{(s)}$ (Eq. (3)), and mapped to a side-wise continuous interaction tolerance radius $\rho_{\text{cont}}^{(s)}$ (Eq. (22)). For single-hand tasks, the pair-level radius is simply the radius of the active side. For bimanual tasks, we aggregate the two active sides and get $\rho_{\text{pair}}(\alpha)$ (Eq. (23)). Because the calibration experiments are available only at the three fixed shared radii 10, 25, and 40 mm, the resulting pair-level radius is projected to the nearest lookup value:

$$\hat{\rho}(\alpha) = \arg \min_{r \in \{10,25,40\}} |r - \rho_{\text{pair}}(\alpha)|. \tag{42}$$

The side-wise computation preserves left/right embodiment differences, whereas the pair-level aggregation matches the shared-radius setting used by the fixed-ρ calibration runs. For calibration pair j, let $A_j(r)$ denote the three-seed mean O-AUC_{1000} obtained with fixed radius r. We define $A_j^{\text{oracle}} = \max_{r \in \{10,25,40\}} A_j(r)$, and the mean calibration regret:

$$\text{Regret}(\alpha) = \frac{1}{12} \sum_{j=1}^{12} \left[A_j^{\text{oracle}} - A_j(\hat{\rho}_j(\alpha)) \right]. \tag{43}$$

We evaluate $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with the calibration results summarized in Table 7. Among the tested candidates, $\alpha = 0.7$ yields the lowest mean O-AUC_{1000} regret and is therefore selected and frozen for all subsequent experiments. Importantly, strict evaluation metrics are not used for this hyperparameter selection. The strict results are evaluated only after calibration and are reported separately in the main experiments.

Table 7: **Calibration on the 12-pair subset.**

α	Projected pairs (10/25/40 mm)	Online AUC regret (pp)
0.1	6/6/0	3.630
0.3	6/6/0	3.630
0.5	4/8/0	2.021
0.7	4/5/3	**1.825**
0.9	1/8/3	2.149

Frozen Task-Hand-Specific Tolerance Mapping. After selecting $\alpha = 0.7$, we no longer project the predicted radii to the three fixed lookup values. Static Compatibility directly generates a task-hand-specific continuous interaction tolerance, which is aggregated across active sides when necessary and rounded to the nearest millimeter. The resulting tolerance is fixed throughout policy learning. Table 8 summarizes the mapped radii used for the 7×4 task-hand matrix.

E Full Task-Wise Curriculum Results

Table 8: **Task-hand-specific SIC tolerance radii with $\alpha = 0.7$.** Values are in mm.

Task	Shadow	Inspire	Artimano	Allegro
Cap the Pen	17	29	16	34
Place Inside	20	33	17	33
Rearrange	24	28	16	33
Put off Alcohol Lamp	17	28	17	30
Uncap the Jar	25	30	26	36
Unscrew	23	28	18	34
Brush Whiteboard	30	36	30	37

Table 9: **Strict comparison of all curriculum variants.** Results are averaged over four hands and three seeds. In each task cell, left / right entries denote S-AUC$_{1000}$ / S-SR$_{E1000}$.

S-AUC$_{1000}$(%) S-SR$_{E1000}$(%)	Cap the Pen (BiH)	Put off Alcohol Lamp (BiH)	Unscrew (BiH)	Uncap the Jar (BiH)	Rearrange (LH)	Place Inside (LH)	Brush Whiteboard (RH)	Average
Baseline	29.6 ± 1.8, 30.9 ± 1.2	44.9 ± 2.0, 56.8 ± 6.5	16.1 ± 1.3, 16.3 ± 2.8	27.3 ± 2.8, 38.9 ± 5.3	30.9 ± 2.1, 24.8 ± 3.8	56.9 ± 1.0, 59.1 ± 1.1	48.5 ± 2.2, 60.3 ± 0.4	36.3, 41.0
No-Curriculum	30.0 ± 2.4, 36.5 ± 0.5	46.5 ± 3.9, 56.2 ± 4.9	17.9 ± 0.5, 19.6 ± 1.0	31.6 ± 0.2, 46.1 ± 2.2	39.4 ± 4.2, 44.5 ± 9.0	57.4 ± 0.2, 58.7 ± 1.3	40.4 ± 0.3, 48.4 ± 0.5	37.6, 44.3
F-TTS	30.3 ± 2.3, 31.2 ± 2.0	42.4 ± 3.8, 55.8 ± 3.5	17.8 ± 0.6, 17.0 ± 1.8	30.7 ± 2.9, 41.3 ± 5.7	37.6 ± 4.6, 39.0 ± 8.0	57.4 ± 1.1, 58.6 ± 2.3	42.2 ± 3.5, 51.6 ± 5.7	36.9, 42.1
DR-RMS	32.0 ± 3.7, 37.3 ± 9.7	47.3 ± 2.7, 60.1 ± 6.0	16.9 ± 0.8, 18.5 ± 0.8	32.6 ± 0.7, 46.5 ± 2.3	38.7 ± 3.2, 39.8 ± 2.9	57.6 ± 0.1, 59.0 ± 1.1	42.3 ± 5.2, 51.7 ± 5.5	38.2, 44.7

F DR-RMS VISUALIZATION

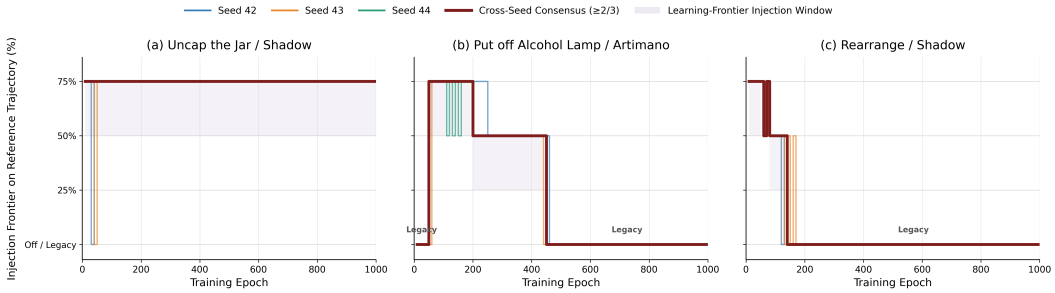


Figure 9: **Evolution of the DR-RMS learning frontier on three representative task-hand pairs.** Thin lines show the learning frontier selected by each training seed, while the thick red line denotes the cross-seed majority state ($\geq 2/3$). The shaded region visualizes the corresponding targeted reset interval: 75%, 50%, and 25% denote injection into the $50-75\%$, $25-50\%$, and $0-25\%$ trajectory segments, respectively; Off/Legacy indicates that no frontier-biased resets are applied and the original ManipTrans sampler is used. The three cases illustrate distinct scheduler behaviors: a persistent frontier for *Uncap the Jar / Shadow*, progressive movement toward earlier phases followed by withdrawal for *Put off Alcohol Lamp / Artimano*, and faster frontier progression and withdrawal for *Rearrange / Shadow*.

G LEARNING CURVES

This section reports the complete strict learning curves over all 28 task-hand pairs for the SC-SIC ablation (Fig. 10), DR-RMS ablation (Fig. 11), and overall CARAT comparison (Fig. 12). Curves show the three-seed mean strict success rate with shaded ± 1 standard deviation.

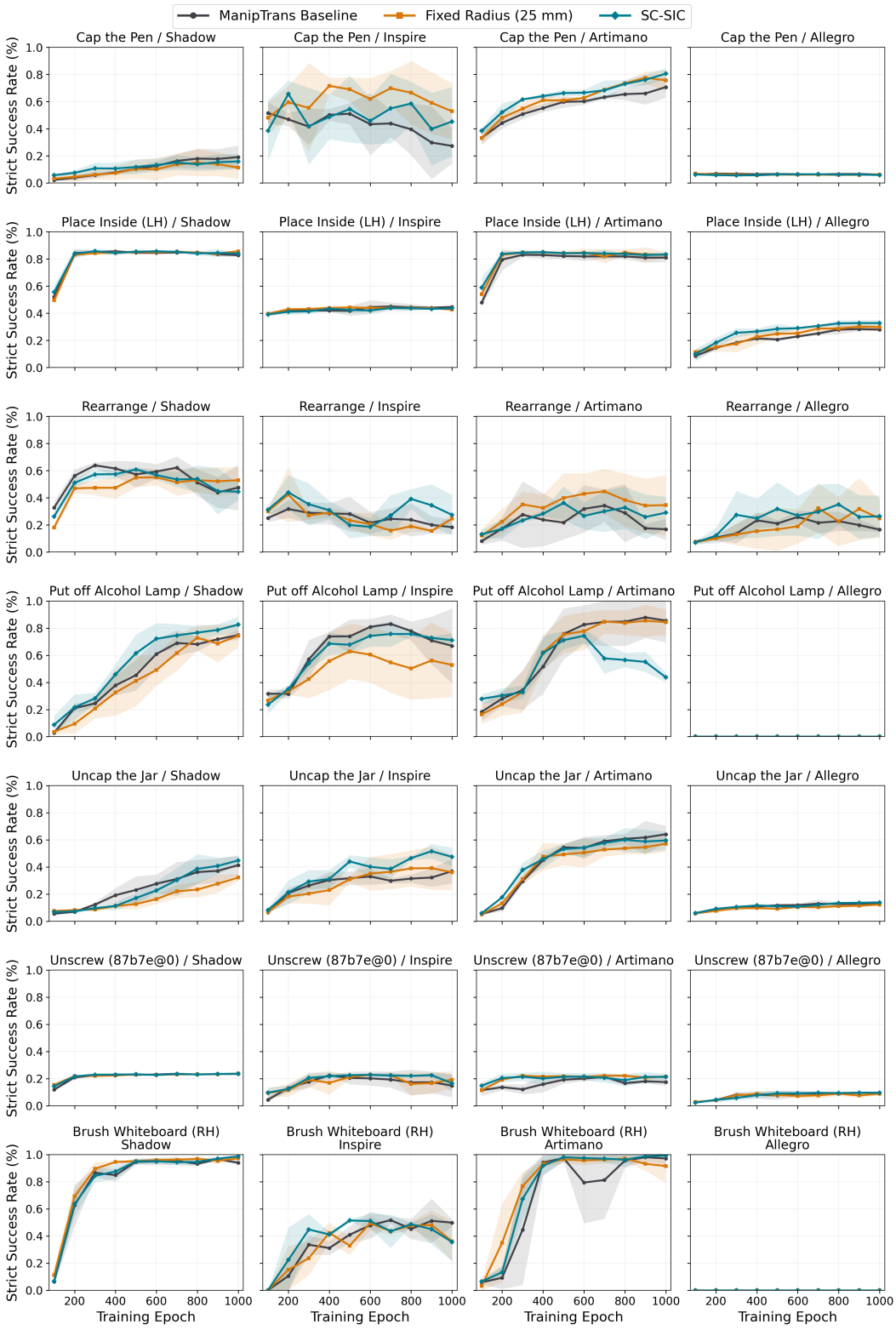


Figure 10: **Strict learning curves for the SC-SIC ablation across all 28 task-hand pairs.** We compare ManipTrans, the globally tuned Fixed-ρ variant ($\rho = 25\,\mathrm{mm}$), and SC-SIC. Curves show the three-seed mean strict success rate, with shaded regions denoting ± 1 standard deviation.

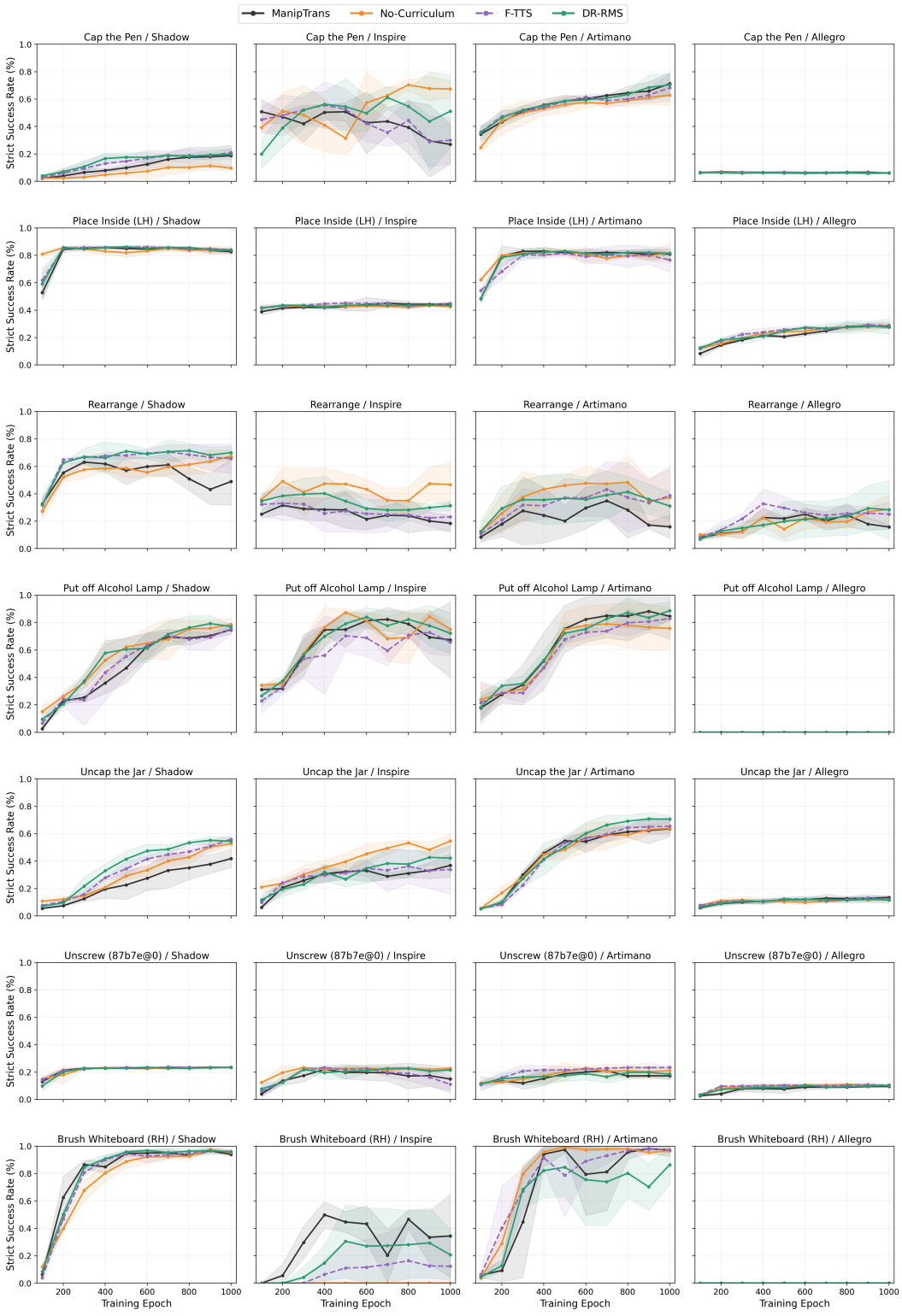


Figure 11: **Strict learning curves for the DR-RMS ablation across all 28 task-hand pairs.** We compare ManipTrans, No-Curriculum, the Fixed Tracking-Threshold Schedule (F-TTS), and DR-RMS. Curves show the three-seed mean strict success rate, with shaded regions denoting ± 1 standard deviation.

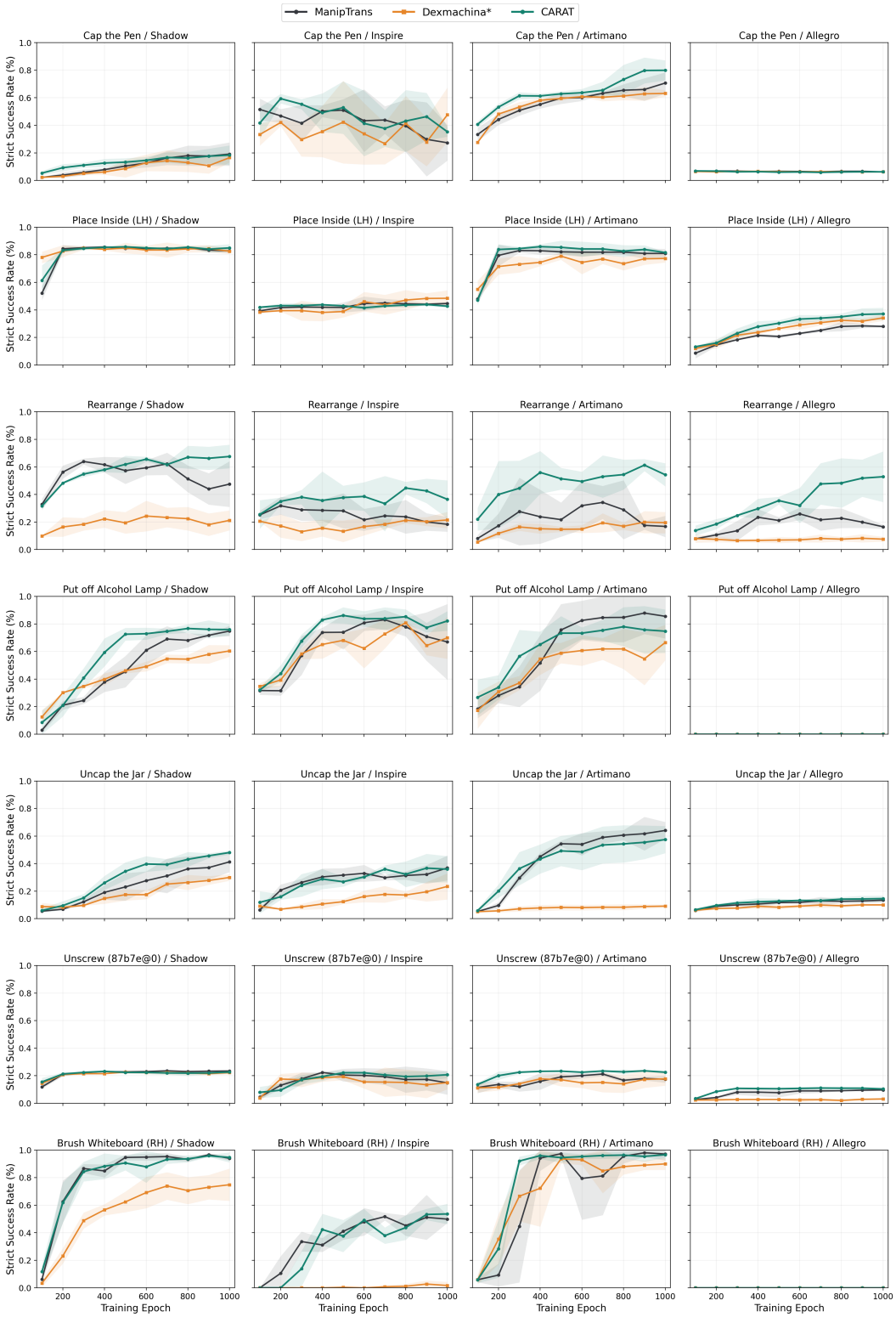


Figure 12: **Strict learning curves for the overall comparison across all 28 task-hand pairs.** We compare ManipTrans, DexMachina*, and the full CARAT framework. Curves show the three-seed mean strict success rate, with shaded regions denoting ± 1 standard deviation.