ROBOMARL-BENCH: HOW TASK STRUCTURE AND DATA COMPOSITION MATTER IN OFFLINE MULTI-AGENT ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline multi-agent reinforcement learning (OMARL) requires agents to learn coordinated behavior from a fixed dataset. Yet task performance and overall dataset quality alone do not reveal how the mix of coordination experiences in the dataset affects what agents can learn. We introduce **RoboMARL-Bench**, a robotics-oriented benchmark with 65 task configurations, 7 rule-defined interaction structures, and 8 controlled data constructions spanning the source, episode, role, temporal, and skill layers. This design enables matched-budget comparisons within the same task while varying how experience is collected and organized. Across extensive evaluations, we find that algorithm rankings change across interaction structures, datasets with the same nominal quality can yield substantially different outcomes, and role assignment and controller ordering introduce additional task-dependent variation. Current OMARL methods also leave clear room for improvement: on data collected by suboptimal policies, they exceed the recorded data level on only about half of the tasks, and most often fall short on multi-goal and transition tasks. RoboMARL-Bench provides a controlled testbed for studying how interaction structure and offline data composition jointly shape multi-agent learning.

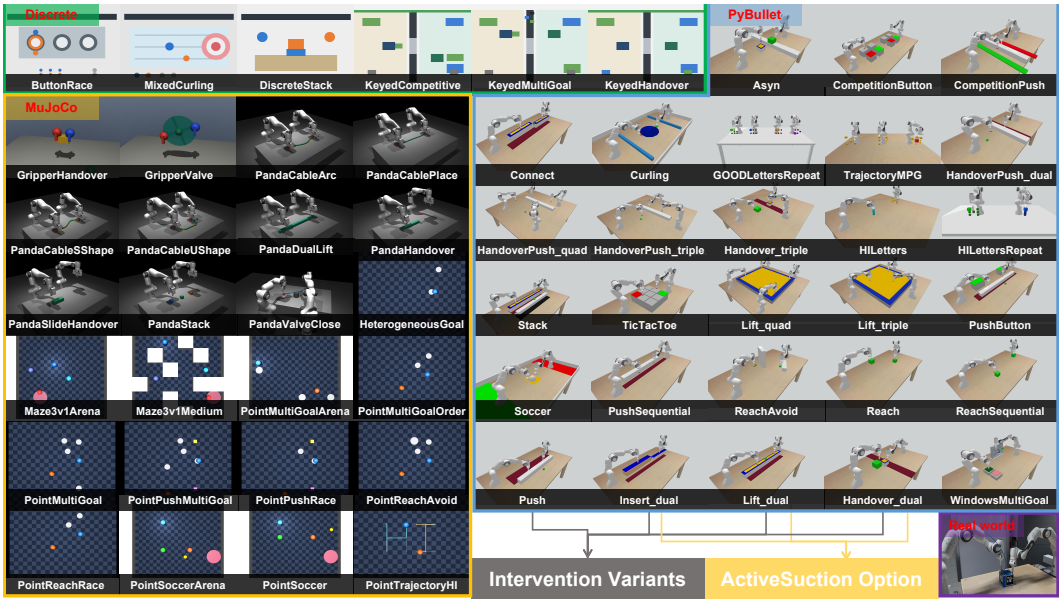


Figure 1: **RoboMARL-Bench Overview.** It comprises 65 registered task configurations: 35 in PyBullet, 24 in MuJoCo, and 6 discrete abstractions. The benchmark provides an active suction interface, a configurable action interface, and inference support for real-world tasks.

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.

1 INTRODUCTION

Offline reinforcement learning (RL) learns from fixed datasets, so performance depends not only on how much data is available but also on how it is collected and composed (Schweighofer et al., 2022; Hong et al., 2023). In multi-agent settings, the recorded experience is further shaped by agent roles and interaction structure (Huh & Mohapatra, 2023; Buşoniu et al., 2010): in robotic manipulation, one robot may determine the state in which another must act, and updating each agent against the joint actions recorded in a dataset can cause coordination failure (Tilbury et al., 2024). Data quality alone may therefore not capture whether the collected experience is suitable for learning coordination.

Existing benchmarks cover standardized offline MARL datasets, multi-robot manipulation, and offline goal-conditioned learning (Formanek et al., 2023; Chernyadev et al., 2024; Chen et al., 2022; Qin et al., 2025; Lai et al., 2025; Wang et al., 2026), while prior work also studies coordination from uncoordinated offline behavior (Ma & Wu, 2023). However, datasets with similar overall quality may differ in which agents contribute high-quality behavior—for example, one agent may follow a random policy while its teammates act competently (Tian et al., 2023)—and in when that behavior occurs within a trajectory. How these differences affect offline learning across interaction structures has received less systematic attention. Studying these effects requires varying data collection while keeping the task and data budget fixed.

We introduce **RoboMARL-Bench** (Figure 1), a robotics-oriented benchmark for offline multi-agent reinforcement learning (OMARL). It contains 65 task configurations, including 59 continuous and 6 discrete tasks, organized into seven rule-defined interaction structures: *Cooperation*, *Competition*, *Multi-Goal*, *Transition*, *Hybrid*, *Trajectory-Tracking*, and *Intervention*. Within the same task, RoboMARL-Bench provides eight data constructions across the **Source**, **Episode**, **Role**, **Temporal**, and **Skill** layers, enabling controlled variation of behavior quality, trajectory mixing, role-wise source assignment, controller ordering, and local skill composition. The benchmark covers diverse learning and evaluation settings, including goal-conditioned learning, restricted observations, image-feature inputs, and real-world dual-arm experiments. Our experiments show that OMARL performance varies with both interaction structure and data composition. Methods that perform strongly on one structure do not consistently dominate others, and datasets assigned the same quality level can lead to substantially different outcomes. Role assignment and controller ordering introduce additional task-dependent variation. Current OMARL methods also leave clear room for improvement: on data collected by suboptimal policies, the two strongest methods exceed the recorded success level on only about half of the tasks, and the tasks where they fall short concentrate in *Multi-Goal* and *Transition*. The related datasets and code will be publicly released. Our contributions are:

- **Explicit interaction structures.** We introduce 65 robot-centric task configurations spanning seven rule-defined interaction structures with task-aligned evaluation metrics, so that methods can be compared per interaction structure rather than through a single aggregate score.
- **Controlled data composition.** We design eight data constructions across the Source, Episode, Role, Temporal, and Skill layers, enabling matched-budget comparisons within the same task and turning how offline data are collected into an explicit experimental variable.
- **Systematic OMARL analysis.** We evaluate how interaction structure and data composition shape OMARL performance, with additional studies on goal-conditioned learning, restricted observations, and physical dual-arm deployment; the results identify suboptimal-policy data and *Multi-Goal* and *Transition* tasks as open problems for future methods.

2 RELATED WORK

Prior work spans three complementary dimensions. For offline data, benchmarks organize datasets by behavior source and quality (Fu et al., 2020) and standardize cooperative offline MARL evaluation (Formanek et al., 2023); related studies show that offline MARL results are tightly coupled to the dataset (Formanek et al., 2024), that demonstration quality and composition shape offline robot

Table 1: Research conditions provided by each benchmark. Compared works are D4RL (Fu et al., 2020), OG-MARL (Formanek et al., 2023), PyMARLzoo+ (Papadopoulos et al., 2025), OGBench (Park et al., 2025), BiGym (Chernyadev et al., 2024), and MangoBench (Wang et al., 2026). A mark means the platform exposes the condition, not that every task carries it.

Research condition	D4RL	OG-MARL	PyMARLzoo+	OGBench	BiGym	MangoBench	RoboMARL-Bench
Multi-arm manipulation[1]	—	—	—	—	40[2]	2	**44** of 65
Agent setting	Single	Multi	Multi	Single	Single	Multi	Both
Learning setting	Offline	Offline	Online	Offline	Online	Offline	Both
Data generation[3]	✓	✗	✓	✓	✓	✗	✓
Interaction taxonomy[4]	✗	✗	✗	✗	✗	✗	✓
Role / timing control[5]	✗	✗	✗	✗	✗	✗	✓
Observability contrast[6]	✗	✗	✗	✗	✗	✗	✓
Standard & GC pool[7]	✗	✗	✗	✓	✗	✗	✓
Real-robot results[8]	✗	✗	✗	✗	✗	✗	✓

[1]Two or more arms acting on a shared scene; a dash means single-arm only. [2]BiGym drives both arms through one unified action interface. [3]Ships collection tooling for new datasets, not only released pools. [4]Organized by reward relations, operational dependencies and completion rules. [5]Role assignment and execution phase varied within one task at fixed budget. [6]Full and partial local state paired on identical trajectories. [7]Standard and goal-conditioned learning over one identical physical pool. [8]Benchmark-trained policies executed on hardware.

manipulation (Mandlekar et al., 2021), and that uncoordinated behavior policies of differing quality impede coordination (Ma & Wu, 2023). For robotic tasks, benchmarks cover demonstration-driven bimanual manipulation (Chernyadev et al., 2024), dexterous multi-agent control (Chen et al., 2022), and compositional multi-robot task generation (Qin et al., 2025). For learning settings, OGBench and MangoBench address offline goal-conditioned learning in single- and multi-agent environments (Park et al., 2025; Wang et al., 2026). RoboMARL-Bench connects these dimensions by varying, within a task and under matched data budgets, how trajectories are mixed, which agent receives which behavior source, and when each controller acts. Table 1 summarizes the comparison.

3 RoboMARL-Bench: Tasks, Formulation, and Evaluation

Tasks. RoboMARL-Bench focuses on multi-arm manipulation and provides 44 robotic-arm tasks: 35 multi-arm tasks in PyBullet (Coumans & Bai, 2016–2021) and 9 Panda-arm tasks in MuJoCo (Todorov et al., 2012). These tasks cover common industrial scenarios, including collaborative lifting, handover, stacking, valve operation, and deformable-object manipulation. The benchmark further includes 13 MuJoCo Point tasks and 2 DualGripper kinematic diagnostic configurations to complement common multi-agent interaction settings, as well as 6 discrete tasks derived from tasks on the two platforms above. In total, it comprises 65 task configurations, 59 with continuous and 6 with discrete actions. Building on these tasks, we define seven rule-defined interaction structures, referred to below as families: *Cooperation*, *Competition*, *Multi-Goal*, *Transition*, *Hybrid*, *Trajectory-Tracking*, and *Intervention* (Figure 2 and Table 27); rules, dimensions, horizons, and success thresholds are listed in Appendix A.24).

Formulation and protocols. We model each task as a finite-horizon partially observable stochastic game $\mathcal{M} = (\mathcal{S}, \{\mathcal{O}_i, \mathcal{A}_i, r_i\}_{i=1}^N, P, \rho_0, \gamma, H)$, in which agent i acts as $a_t^i \sim \pi_i(\cdot \mid o_t^i, g_t^i)$ given its local observation o_t^i and goal g_t^i. For robotic manipulation tasks, we primarily adopt the RELATIVE configuration, in which observations are expressed in local coordinate frames: each agent retains information about neighboring roles and its own joint state, and its goal is also expressed in the local frame rather than in absolute coordinates. For tasks involving multiple objects and for *Transition* tasks, we further provide a PARTIAL configuration, in which the arm associated with each agent observes only its own region. On top of the PARTIAL configuration, the FEATURES configuration concatenates the output of a fixed image encoder with valid proprioceptive or local-state information and the goal, $x_t^i = [f_\theta(I_t^i), p_t^i, g_t^i]$. We also support a goal-conditioned (GC) protocol, in which valid future goals are sampled within the same episode and each agent's local goal representation, reward, and bootstrap mask are recomputed accordingly; failure termination, budget truncation, and goal achievement are handled separately. Exact observation inputs, goal-relabeling procedures, and baseline configurations are specified in Appendix A.23.

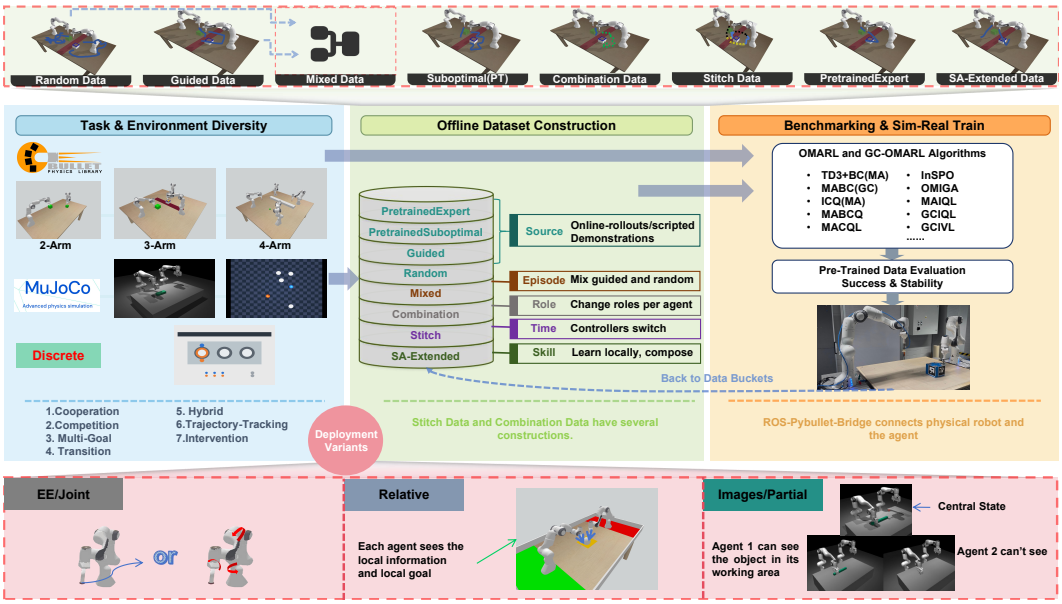


Figure 2: **RoboMARL-Bench workflow.** The benchmark integrates task-structure and dataset-composition analysis, algorithm evaluation, and real-world policy inference. PT is short for pre-trained.

Evaluation. For *Cooperation*, *Transition*, *Multi-Goal*, *Hybrid*, and *Intervention* tasks, we report the episode success rate. For *Trajectory-Tracking* tasks, we report the fraction of episodes that complete at least one full pattern (also called the success rate) for the two one-shot tasks, and the number of completed patterns per episode for the three repeated tasks. For close-proximity dual-arm tasks, we additionally report the inter-arm clearance-violation rate as a reference metric (Appendix A.15). For *Competition* tasks, we evaluate the policy separately on each side or team against a fixed random opponent (Appendix A.11 examines a fixed learned opponent). We denote by W_i the win rate when the policy controls side i, and by $W_{\min}$ the win rate of the weaker side (Appendix A.10):

$$W_i = \Pr(\pi_i \text{ wins when controlling side } i)\,, \qquad W_{\min} = \min_i W_i. \tag{1}$$

4 Offline Data Construction and Interaction Evidence

4.1 Three Quality Levels, Five Layers, and Eight Data Constructions

We organize the data into three quality levels, five layers, and eight data constructions. The three quality levels are **Good**, **Medium**, and **Poor**, and the five layers are **Source**, **Episode**, **Role**, **Temporal**, and **Skill**. At the **Source** layer, we include *PretrainedExpert*, *PretrainedSuboptimal*, *Guided*, and *Random*. The first two are generated from high-quality and suboptimal online checkpoints using online-MASAC (Pu et al., 2021), respectively; *Guided* uses a scripted generator, while *Random* executes valid random actions. At the **Episode** layer, *Mixed* combines complete *Guided* and *Random* episodes according to the transition budget. At the **Role** layer, *Combination* assigns controllers from different sources to different agents. At the **Temporal** layer, *Stitch* switches controllers within a single physical episode without resetting the environment state. At the **Skill** layer, *SA-Extended* first learns local policies with fixed scripted partners and then collects data through joint execution. During data collection, we additionally perturb the actions (Appendix A.22). In the main experiments, **Good** uses *PretrainedExpert* when a qualified checkpoint is available and *Guided* otherwise; **Medium** uses *PretrainedSuboptimal* under the same rule and *Mixed* otherwise; **Poor** uses *Random*. Appendix A.22 details the constructions, and Appendix A.20 characterizes the resulting pools by reference-relative coverage, showing that the support of a pool cannot be summarized by its quality label or action diversity alone.

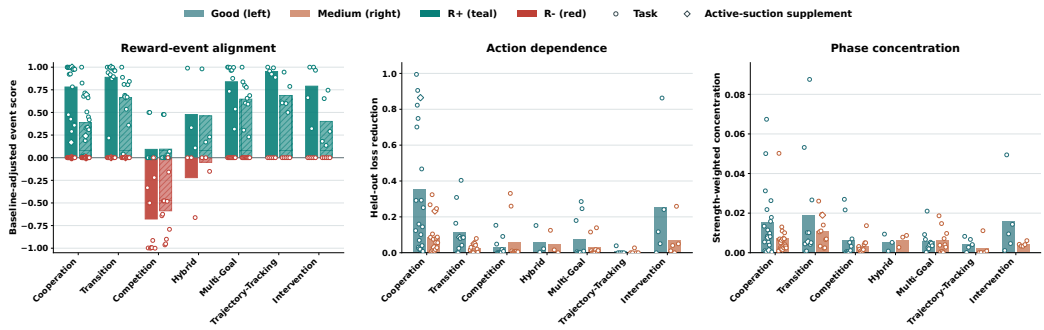


Figure 3: **Interaction diagnostics for all 65 configurations in Good and Medium.** Bars are task-equal family means, Good left and Medium hatched right; circles are tasks, and diamonds the three active-suction variants. Baseline-adjusted R^+ (teal) and R^- (red) sit above and below zero; C_A and C_P use episode-disjoint held-out data. They describe sampled behavior, not family membership.

4.2 Structural Signatures in Collected Behavior

Families are defined by environment rules. We further examine whether they exhibit distinguishable empirical signatures in collected behavior, without attributing source-dependent differences to the family itself. For each physical data pool, we compute three descriptive diagnostics: reward-event alignment $R^{\pm}$, which measures whether non-baseline reward events across agents are aligned or opposing; conditional action dependence C_A, which measures the incremental predictive information provided by a peer's current action given the state, local observation, and goal; and phase concentration C_P, which measures whether such action dependence is concentrated in particular execution stages. Full definitions, estimators, and missingness rules are provided in Appendix A.4. The central state is used only for these post-hoc diagnostics and is never provided to decentralized actors. As Figure 3 shows, each diagnostic separates a different family: reward-event alignment discriminates *Competition* most sharply, action dependence does so for *Cooperation*, and phase concentration is highest for *Transition*. No single quantity orders all seven families, so the three diagnostics are complementary.

4.3 Task-Matched Response to Data Composition

To test whether dataset constructions are interchangeable within a fixed task, we define six contrasts over the 14 continuous tasks selected for the matched-budget comparison (Section 5.2), each altering one aspect of the data: episode composition (*Mixed*−*Guided*), role assignment ($|\mathrm{E/S} - \mathrm{S/E}|$, computed on the 13 two-agent tasks), two controller orderings ($(\mathrm{R} \to \mathrm{G}) - (\mathrm{G} \to \mathrm{R})$ and $(\mathrm{R} \to \mathrm{E}) - (\mathrm{E} \to \mathrm{R})$), behavior-source quality as the reference axis ($\mathrm{S} - \mathrm{E}$), and local information as a separate one (Relative−Partial), which is computed on 14 of the 16 observation-comparison tasks (Section 5.3). E and S denote the high-quality and suboptimal online sources (*Pretrained-Expert*, *PretrainedSuboptimal*), and G and R denote *Guided* and *Random*. All six contrasts share one aggregation rule and differ only in the pair (c_1, c_2) substituted into it. Writing $\bar{J}_e^A(c)$ for the cross-seed mean Final performance of task e under method A and condition c, we difference within each method before averaging methods with equal weight to obtain the task response, and take the within-family median as the family response:

$$\delta_c(e) = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} \left[\bar{J}_e^A(c_1) - \bar{J}_e^A(c_2) \right], \qquad \Delta_c(F) = \operatorname*{median}_{e \in F} \delta_c(e). \tag{2}$$

Role assignment is the one exception: we take $|\bar{J}_e^A(\mathrm{E/S}) - \bar{J}_e^A(\mathrm{S/E})|$ within each method before averaging, because the favored direction varies across tasks and signed averaging would cancel it to zero.

Under equal transition budgets, the constructions are not interchangeable: *Mixed* trails *Guided* on every source-study task, handing control to the purposeful controller last is better on nearly every task (the few exceptions are *Competition* tasks), and role swaps matter more in *Transition* than in

Cooperation or *Multi-Goal* (Appendix A.5). These comparisons characterize differences between complete collection procedures rather than the causal effect of any single factor.

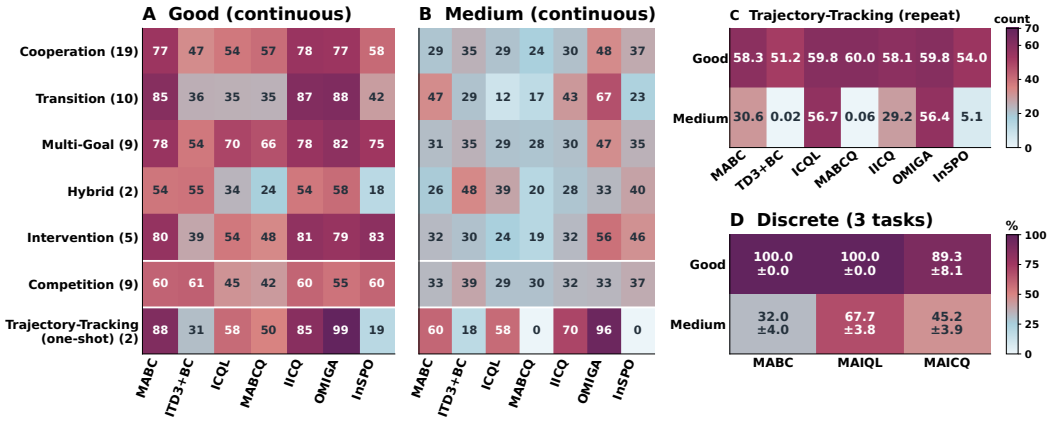


Figure 4: **Method performance across interaction structures.** (A, B) Final results under Good and Medium data for seven continuous-action methods under the Independent configuration on the same 56 percentage-valued tasks. Cells report five-seed, task-equal family means. *Competition* uses the weaker-side win rate $W_{\min}$, and the two one-shot *Trajectory-Tracking* tasks use episode completion. (C) The three repeated *Trajectory-Tracking* tasks, in completed patterns per episode. (D) Three cooperative discrete tasks, with five-seed sample standard deviations. The lower color scale applies to A, B, and D (0–100%), and the upper scale to C (0–70 patterns per episode). MuJoCo Panda/Point tasks use 500k transitions and other continuous tasks use 50k; all methods are trained for 50k updates. Per-task results are in Appendix A.6, and family-level sample standard deviations in Appendix A.8.

5 Experiments

Our experiments address two questions: (1) **how do different algorithms perform across interaction structures?** and (2) **how does performance change within the same task when the data source, role assignment, or execution order is varied?** Unless marked as Best, all results use the Final checkpoint at the end of a fixed training budget (Appendix A.16), and each checkpoint is evaluated independently for 100 episodes. Unless stated otherwise, training uses 50k updates, a batch size of 512, and five seeds. For continuous-action tasks, the main comparison includes MABC (Torabi et al., 2018), ITD3+BC (Fujimoto & Gu, 2021), ICQL (Kumar et al., 2020), MABCQ (Jiang & Lu, 2023), IICQ (Yang et al., 2021), OMIGA (Wang et al., 2023), and InSPO (Liu et al., 2024) under the Independent configuration (Appendix A.18). Discrete tasks use MABC, MAIQL (Kostrikov et al., 2021), and MAICQ (Yang et al., 2021), while goal-conditioned experiments use GCBC, GCIQL (Kostrikov et al., 2021), and GCIVL (Kostrikov et al., 2021; Park et al., 2024). We write pp for percentage points.

5.1 Performance Across Interaction Structures

Figure 4 compares the overall performance of different algorithms across interaction structures under Good and Medium data. Panels A–C cover all 59 continuous-action tasks, evaluated by success rate or completed patterns, while Panel D reports results on discrete-action tasks. For continuous-action tasks, OMIGA performs well overall under Good data, but its margin is small: in most families the leading methods lie within a few points of each other. Under Medium data the gap widens, and OMIGA leads clearly on *Cooperation*, *Transition*, *Multi-Goal*, *Intervention*, and one-shot *Trajectory-Tracking* tasks. It remains weaker on *Competition* tasks under both quality levels and on the two *Hybrid* tasks under Medium data, where ITD3+BC scores highest, so no single method leads in every family (Appendix A.12). On the repeated *Trajectory-Tracking* tasks, all methods perform similarly under Good data; under Medium data, ICQL and OMIGA retain most of their performance, whereas ITD3+BC, MABCQ, and InSPO degrade to near zero. For discrete-action tasks, MABC and MAIQL both reach 100% under Good data, and MAIQL achieves the highest

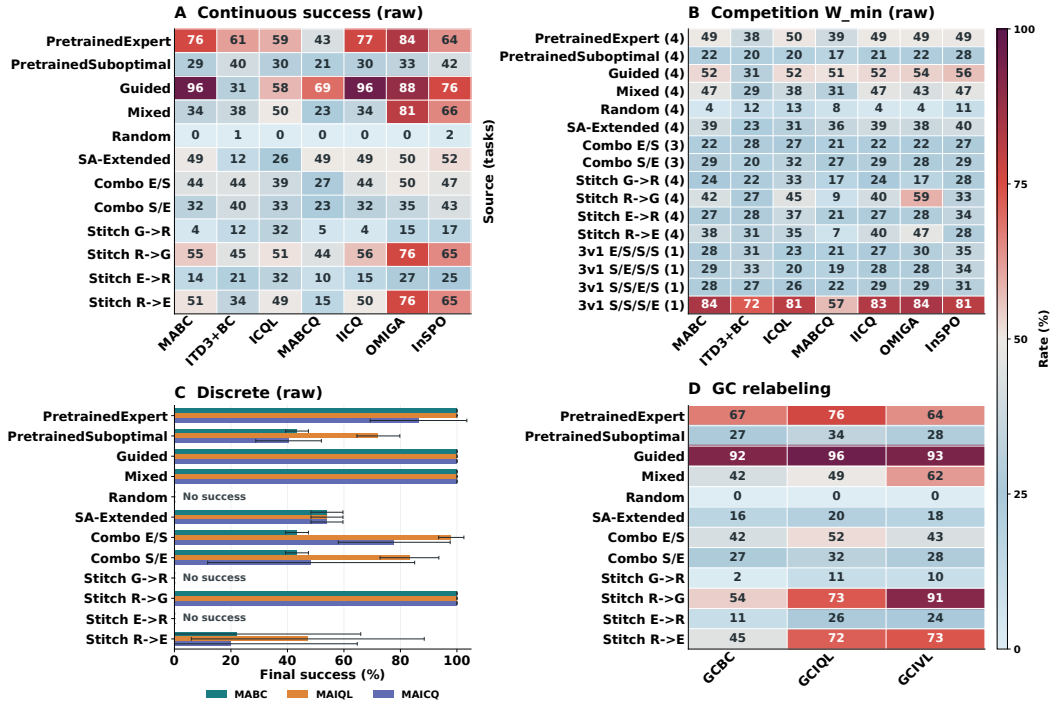


Figure 5: **Data-composition results at matched budgets.** (A) Final performance on nine continuous tasks and one one-shot *Trajectory-Tracking* task (completion $\times 100$) for seven Independent methods. (B) Four *Competition* tasks, $W_{\min}$ against fixed random opponents; parentheses give task counts. (C) Discrete Stack, MABC/MAIQL/MAICQ, with five-seed sample standard deviations; "No success" marks zero for all three. (D) Goal-conditioned, four tasks, GCBC/GCIQL/GCIVL. All pools hold 50k transitions; training uses 50k updates and five seeds. Standard deviations for A, B, and D are in Appendix A.8.

performance under Medium data (68%). We report the three discrete tasks that allow a common comparison across all methods; MAICQ does not apply to the discrete *Competition* tasks.

5.2 Matched-Budget Data Constructions and Goal-Conditioned Learning

To further study how data sources affect task performance, and to systematically compare three aspects of data composition—source quality, role assignment, and temporal ordering—we select a representative set of tasks with the corresponding data constructions, together with Discrete Stack as an additional analysis at the discrete-control level. The PyBullet tasks are Insert, Connect, Lift, Handover, ReachSequential, Curling, PushIntervention, CompetitionButton, and CompetitionPush; the MuJoCo tasks are PandaHandover, PointMultiGoalArena, PointMaze3v1Arena, PointReachRace, and PointTrajectoryHI. We use abbreviated task names throughout; the full names are provided in Appendix A.24. These tasks cover all seven families. We do not extend this analysis to every task because high-quality expert checkpoints are difficult to obtain through online training for some tasks. To ensure comparability across data constructions, all continuous tasks in this analysis use 50k transitions, including the MuJoCo tasks; in the remaining main experiments, MuJoCo tasks use 500k transitions (Appendix A.17).

As shown in Figure 5, a data-quality label alone does not fully characterize how methods respond to different data compositions. Panel A reports continuous non-competitive tasks.OMIGA reaches 33.3% on *PretrainedSuboptimal* but 80.8% on *Mixed*, although both are labeled **Medium**; the recorded success rates of these pools (32.1% vs. 69.5%) show that one quality label can cover very different data. Controller ordering within joint trajectories has a similarly large effect: under the *Random*→*Guided* Stitch construction, OMIGA reaches 76.5%, compared with only 15.4% for the reverse *Guided*→*Random* ordering. Panel B reports *Competition* tasks separately. Because $W_{\min}$ is measured against a random opponent, a policy trained on *Random* data can still win some games

by chance, so the *Random* row in Panel B is not comparable to that in Panel A. Role assignment, examined through the *Combination* construction, matters most clearly in the 3-vs-1 task: across methods, $W_{\min}$ stays at 19–35% when any of the first three seats receives *PretrainedExpert* data, but rises to 57–84% when the fourth seat, the single-agent side, receives it (S/S/S/E). Similar composition effects appear under goal-conditioned training (Panel D; Appendix A.7), where future goals are relabeled within episodes for 48 physical 50k-transition pools across four tasks, with local goals, rewards, and bootstrap masks updated accordingly. GCIQL achieves 52.1% and 31.7% under E/S and S/E, respectively, while GCIVL reaches 91.3% under *Random→Guided* and 10.4% under the reverse ordering.

5.3 Observation Constraints and Two References for Learning Gain

RoboMARL-Bench provides PARTIAL and FEATURES interfaces for 16 tasks, 14 in PyBullet and 2 in MuJoCo (Table 25). As shown in Figure 6, we compare seven algorithms on these 16 tasks. Each data pool contains 50k transitions, and all three configurations share four training seeds; RELATIVE and PARTIAL are trained for 50k updates, and FEATURES for 100k. Overall, task difficulty increases from RELATIVE to PARTIAL and then to FEATURES, with a corresponding decrease in average success rate. For example, under Guided data, OMIGA decreases from 79.5% in the RELATIVE configuration to 72.1% in PARTIAL. However, FEATURES is not uniformly the most difficult setting: under Mixed data, ICQL reaches 38.1% with FEATURES, compared with 16.7% and 17.3% under RELATIVE and PARTIAL. The effect of observation mode is thus strongly method-dependent.

We next examine whether OMARL can extract learning gains beyond the performance level recorded in the offline data. To keep a single metric, we exclude the 9 *Competition* tasks, the 5 *Trajectory-Tracking* tasks, and the 6 discrete tasks, leaving 45 continuous tasks whose native metric is success rate. Because high-quality offline data typically leads to strong policy performance, Medium data provides a more informative setting for assessing whether OMARL can exploit limited-quality data. We therefore use the Medium split for all 45 tasks, of which 28 are sourced from *PretrainedSuboptimal* and 17 from *Mixed*.

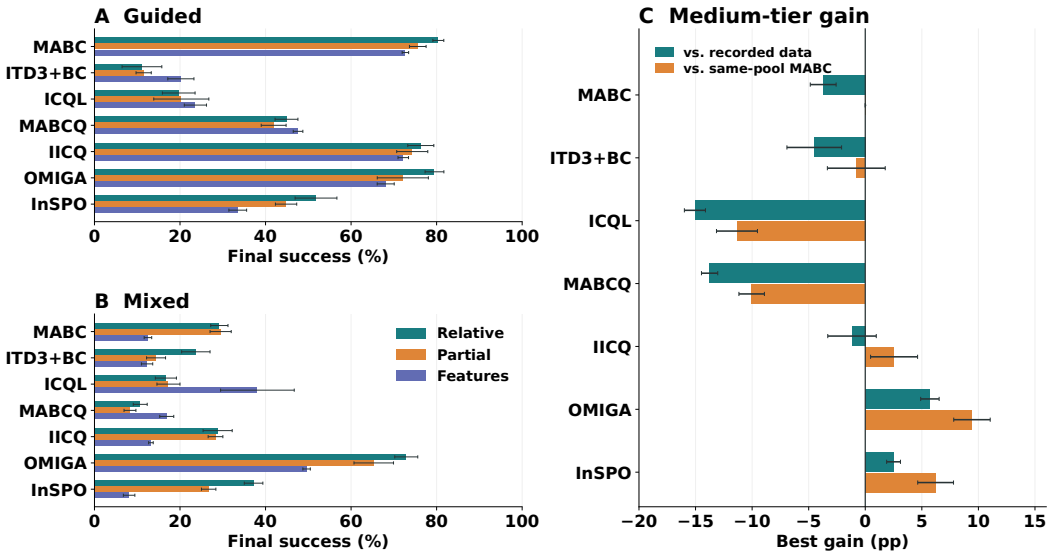


Figure 6: **Observation comparison and learning gain.** (A, B) Final success on the 16 observation-comparison tasks with the Guided (A) and Mixed (B) pools (50k transitions, four common seeds). RELATIVE and PARTIAL are trained for 50k updates and FEATURES for 100k, so gaps involving FEATURES also reflect the training budget. (C) Best-checkpoint gain on the 45 Medium tasks over five seeds, measured against the recorded success rate (Δ^{data}) and same-pool MABC (Δ^{BC}). Error bars are sample standard deviations of seed-level task means. Panel C differs from A and B in task set, checkpoint rule, and training budget, so values are not comparable across panels.

Table 2: **Real-robot successes out of 30 trials per task and method.** Connect-Insert keeps the distance-based completion criterion of the simulated task, so trials that meet it without full insertion also count as successes (Appendix A.1).

Task	MABC	OMIGA	InSPO
Reach	30/30	30/30	30/30
Lift	30/30	23/30	24/30
Handover (PARTIAL)	11/30	17/30	18/30
Connect-Insert (PARTIAL)	11/30	16/30	18/30

To distinguish improvement beyond the recorded dataset level from improvement beyond direct imitation, let q_e denote the recorded success fraction of retained episodes for task e within the same H5 dataset, and let $J_{e,s}(A)$ denote the independently evaluated success rate of method A under training seed s. We define two percentage-point gaps:

$$\Delta^{\mathrm{data}}_{e,s}(A) = J_{e,s}(A) - 100\, q_e, \qquad \Delta^{\mathrm{BC}}_{e,s}(A) = J_{e,s}(A) - J_{e,s}(\mathrm{MABC}). \tag{3}$$

Here, Δ^{data} measures the gain over the success rate recorded in the dataset, and Δ^{BC} the gain over MABC trained on the same pool, evaluation protocol, and paired seed. Because q_e is computed from the selected pool, which may include exploratory actions, and *Mixed* pools combine complete *Guided* and *Random* episodes, a test success rate above $100q_e$ does not imply superiority over the pool's best constituent behavior. We therefore report *PretrainedSuboptimal* and *Mixed* separately.

Across the 45 Medium tasks, the task-equal recorded success rate is 47.59%. OMIGA and InSPO achieve Best (validation-selected) performance of 53.30% and 50.09%, corresponding to gains of 5.71 pp and 2.50 pp over the recorded level, and exceed the recorded success rate on $30/45$ and $25/45$ tasks, respectively. Relative to MABC Best, their mean gains are 9.43 pp and 6.22 pp. Split by source, OMIGA exceeds the recorded level on $15/28$ *PretrainedSuboptimal* and $15/17$ *Mixed* tasks, and InSPO on $14/28$ and $11/17$: on the stricter *PretrainedSuboptimal* pools, both methods improve on the recorded behavior for only about half of the tasks. Tasks that do not exceed the recorded level concentrate in MuJoCo *Multi-Goal* and *Transition* manipulation tasks, where current OMARL methods leave clear room for improvement.

5.4 REAL-ROBOT VALIDATION

To show that policies trained on the benchmark transfer to hardware, we connect the simulator to two Franka Panda arms through a ROS–PyBullet bridge (Mower et al., 2022; Quigley et al., 2009) and deploy MABC, OMIGA, and InSPO on four dual-arm tasks: Reach, Lift, and the Partial-observation variants of Handover and Connect-Insert. Each method–task pair is run for 30 trials (Table 2, Appendix A.1). The experiment demonstrates that the platform extends to real hardware rather than re-testing the data-composition findings; the same interface also offers a route for recording real-world episodes in the benchmark's data format, which we leave to future work.

6 DISCUSSION

Our analysis is descriptive rather than causal. The interaction families are defined by environment rules but are not isolated from embodiment, horizon, or action space, and some contain few tasks; comparisons within the same operation still show method-dependent responses (Appendix A.9). Changing role assignment or controller order alters the whole trajectory distribution rather than role or timing alone, and the real-robot study tests deployability rather than the data-composition findings. Within these limits and the documented method adaptations, RoboMARL-Bench shows that OMARL performance varies both with *what interaction a task requires* and with *how offline experience is composed*: no single method leads in every family, datasets with the same quality label can differ substantially in content and outcome, and role assignment and controller ordering add task-dependent variation. Current methods also leave clear room for improvement on several *Multi-Goal* and *Transition* tasks. The benchmark thus aims not at a single algorithm ranking but at reproducible conditions for studying which forms of offline experience support which forms of multi-agent interaction.

AI USE STATEMENT

All scientific content of this paper is our own. The benchmark tasks, the data-construction scheme, the collection and training pipelines, the experiments, and the analysis of their results were conceived and carried out by the authors, and no dataset, hypothesis, or method implementation was generated by an AI system. Large language models played a supporting role in four respects:

1. **Writing.** Improving the wording of sections the authors had already drafted, including translating parts of the drafts into academic English.
2. **Figures and tables.** Adjusting the layout and styling of result figures and LaTeX tables. Every plotted or tabulated value comes from our own evaluation logs, and the regenerated figures were checked against those logs.
3. **Code.** Helping revise code written by the authors. Every such change was reviewed and re-tested by the authors before any experiment relied on it.
4. **Result analysis.** We analyzed the experimental results ourselves and then compared our reading with one produced by an LLM, to catch anything we might have missed. Where the two disagreed, we went back to the evaluation logs.

The paper contains no theoretical results, so assistance with proofs or mathematical claims does not arise. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe all 65 task configurations in Appendix A.24, including observation and action dimensions, horizons, and success criteria; the environment interface is given in Appendix A.2. Appendix A.22 explains how the data for each of the eight constructions are collected, with the action perturbations and transition budgets used. Appendices A.23 and A.23 list observation inputs, goal relabeling, and the training settings for every baseline, and Appendix A.18 describes our changes to OMIGA and InSPO for the Independent configuration. Unless noted otherwise, each result is averaged over five training seeds, and each checkpoint is evaluated on 100 episodes. Per-task results are in Appendix A.6 and seed-level variability in Appendix A.8. The supplementary material provides the anonymized code for all experiments (environments, data collection, training, and evaluation) and sample datasets. The full datasets are too large for the supplementary material; we will release them publicly with the code.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay, 2018. URL https://arxiv.org/abs/1707.01495.

Paul Barde, Jakob Foerster, Derek Nowrouzezahrai, and Amy Zhang. A model-based solution to the offline multi-agent reinforcement learning coordination problem, 2024. URL https://arxiv.org/abs/2305.17198.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuang Jiang, Stephen Marcus McAleer, Yiran Geng, Hao Dong, Zongqing Lu, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning, 2022. URL https://arxiv.org/abs/2206.08686.

Nikita Chernyadev, Nicholas Backshall, Xiao Ma, Yunfan Lu, Younggyo Seo, and Stephen James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. *arXiv preprint arXiv:2407.07788*, 2024.

David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study, 2014. URL https://arxiv.org/abs/1404.3785.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

Claude Formanek, Asad Jeewa, Jonathan Shock, and Arnu Pretorius. Off-the-grid marl: Datasets with baselines for offline multi-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2302.00521.

Claude Formanek, Louise Beyers, Callum Rhys Tilbury, Jonathan P. Shock, and Arnu Pretorius. Putting data at the centre of offline multi-agent reinforcement learning, 2024. URL https://arxiv.org/abs/2409.12001.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2014.01.005. URL https://www.sciencedirect.com/science/article/pii/S0031320314000235.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

Zhang-Wei Hong, Pulkit Agrawal, Rémi Tachet des Combes, and Romain Laroche. Harnessing mixed offline reinforcement learning datasets via trajectory weighting. *arXiv preprint arXiv:2306.13085*, 2023.

Dom Huh and Prasant Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

Jiechuan Jiang and Zongqing Lu. Offline decentralized multi-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2108.01832.

Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 374(2065):20150202, 2016.

Maurice G Kendall et al. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning, 2020. URL https://arxiv.org/abs/2006.04779.

Matthew Lai, Keegan Go, Zhibin Li, Torsten Kröger, Stefan Schaal, Kelsey Allen, and Jonathan Scholz. Roboballet: Planning for multirobot reaching with graph neural networks and reinforcement learning. *Science Robotics*, 10(106):eads1204, 2025.

Zongkai Liu, Qian Lin, Chao Yu, Xiawei Wu, Yile Liang, Donghui Li, and Xuetao Ding. Offline multi-agent reinforcement learning via in-sample sequential policy optimization, 2024. URL https://arxiv.org/abs/2412.07639.

Jinming Ma and Feng Wu. Learning to coordinate from offline datasets with uncoordinated behavior policies. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023. URL https://aamas.csc.liv.ac.uk/Proceedings/aamas2023/pdfs/p1258.pdf.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021. URL https://arxiv.org/abs/2108.03298.

Christopher E. Mower, Theodoros Stouraitis, João Moura, Christian Rauch, Lei Yan, Nazanin Zamani Behabadi, Michael Gienger, Tom Vercauteren, Christos Bergeles, and Sethu Vijayakumar. Ros-pybullet interface: A framework for reliable contact simulation and human-robot interaction, 2022. URL https://arxiv.org/abs/2210.06887.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification, 2022. URL https://arxiv.org/abs/2111.11188.

George Papadopoulos, Andreas Kontogiannis, Foteini Papadopoulou, Chaido Poulianou, Ioannis Koumentis, and George Vouros. An extended benchmarking of multi-agent reinforcement learning algorithms in complex fully cooperative tasks, 2025. URL https://arxiv.org/abs/2502.04773.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions, 2024. URL https://arxiv.org/abs/2307.11949.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl, 2025. URL https://arxiv.org/abs/2410.20092.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018.

Yuan Pu, Shaochen Wang, Rui Yang, Xin Yao, and Bin Li. Decomposed soft actor-critic method for cooperative multi-agent reinforcement learning, 2021. URL https://arxiv.org/abs/2104.06655.

Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints, 2025. URL https://arxiv.org/abs/2503.16408.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, pp. 5. Kobe, Japan, 2009.

Kajetan Schweighofer, Marius-constantin Dinu, Andreas Radler, Markus Hofmarcher, Vihang Prakash Patil, Angela Bitto-Nemling, Hamid Eghbal-Zadeh, and Sepp Hochreiter. A dataset perspective on offline reinforcement learning. In *Conference on Lifelong Learning Agents*, pp. 470–517. PMLR, 2022.

Qi Tian, Kun Kuang, Furui Liu, and Baoxiang Wang. Learning from good trajectories in offline multi-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2211.15612.

Callum Rhys Tilbury, Claude Formanek, Louise Beyers, Jonathan P. Shock, and Arnu Pretorius. Coordination failure in cooperative offline marl, 2024. URL https://arxiv.org/abs/2407.01343.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation, 2018. URL https://arxiv.org/abs/1805.01954.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization, 2023. URL https://arxiv.org/abs/2307.11620.

Yi Wang, Ningze Zhong, Zhiheng Fu, Longguang Wang, Ye Zhang, and Yulan Guo. Mangobench: A benchmark for multi-agent goal-conditioned offline reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6219–6228, 2026.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning, 2021. URL https://arxiv.org/abs/2106.03400.

A Appendix

A.1 Real-world Implementation

As shown in Figure 7, we use techniques such as domain randomization to support the sim-to-real transfer process.

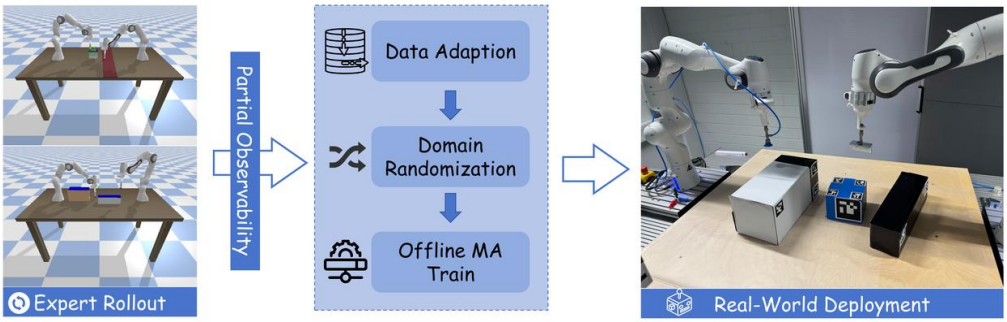


Figure 7: **The overall pipeline**. Expert data can be collected through this platform (either from single-agent or multi-agent settings). The data is then reformulated to fit the partially observable setting, followed by domain randomization. Finally, OMARL training is conducted, and the learned policies are deployed in the real world.

In the real-world experiments, we use two Franka Emika Panda robotic arms, as shown in Figure 8 and Figure 9. We employ two independent Intel RealSense D435i and Intel RealSense D435 cameras to provide pose information (combined with ArUco markers (Garrido-Jurado et al., 2014)). The figure illustrates all objects used in the tasks and the placement of the cameras.

As mentioned earlier, we adopt a ROS–PyBullet bridge to transfer information between the simulation and the real robotic system. In simulation, the agent operates under end-effector control. Through the bridge, the desired joint targets computed in simulation are sent to the real robotic arms and executed through trajectory planning using the MoveIt framework (Coleman et al., 2014).

However, there are configuration differences between the real robot and its simulated counterpart, including joint limits, response speed, and trajectory planning behavior. In particular, the real-world system relies on the MoveIt planner, whereas the simulator uses PyBullet's built-in control and planning mechanisms. Although the commands generated by the agent remain the same, different planners may lead to different motion behaviors. For example, one planner may generate a detour to reach the same target pose, while another may move more directly from point to point.

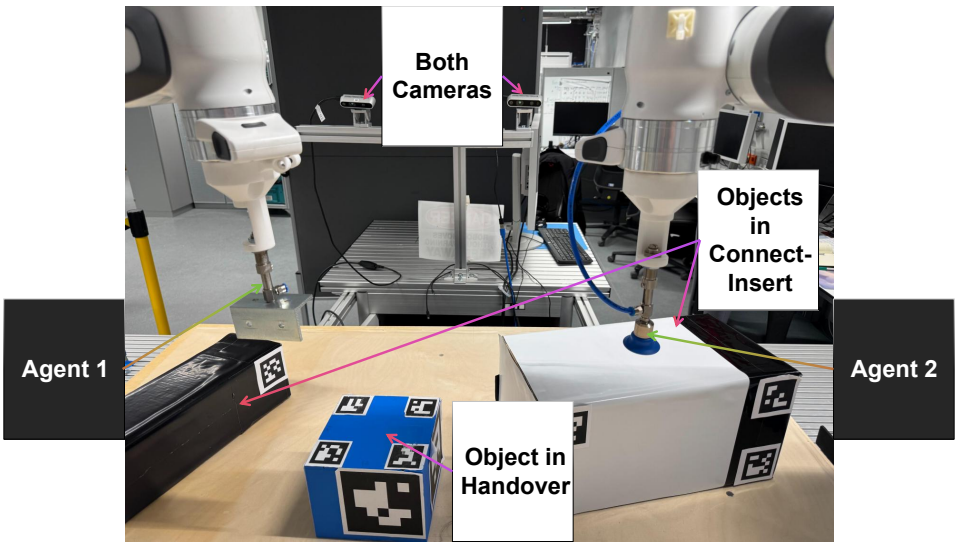


Figure 8: **The real-world setups.** Here we include all objects used in these two tasks, and the two cameras can only see partial information about the objects.

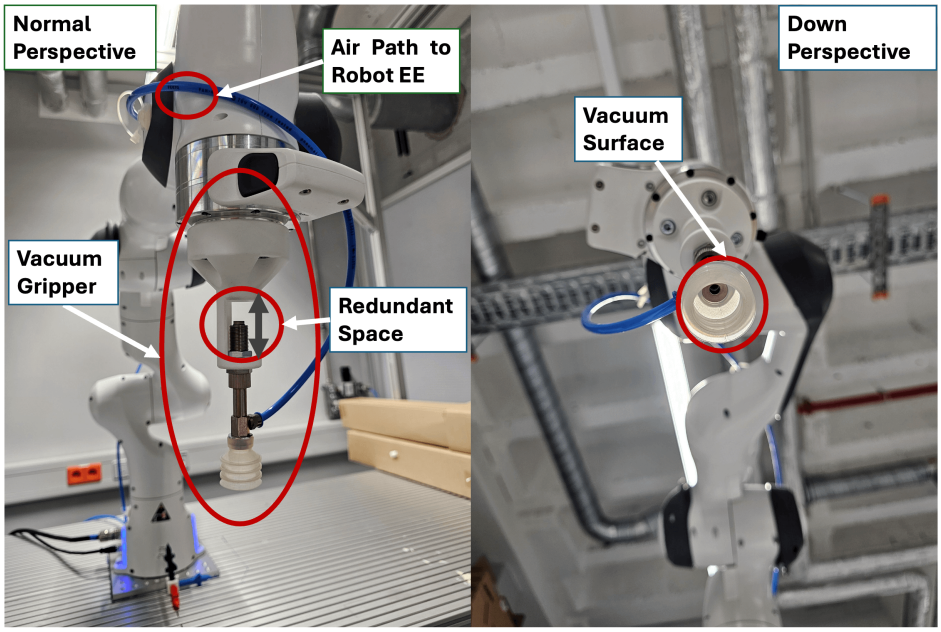


Figure 9: **Vacuum Gripper Setups.** Here, we provide both a normal view and an underside view, with key details highlighted. For example, the gripper features a redundant vertical movement design, which better aids in the sim-to-real transfer.

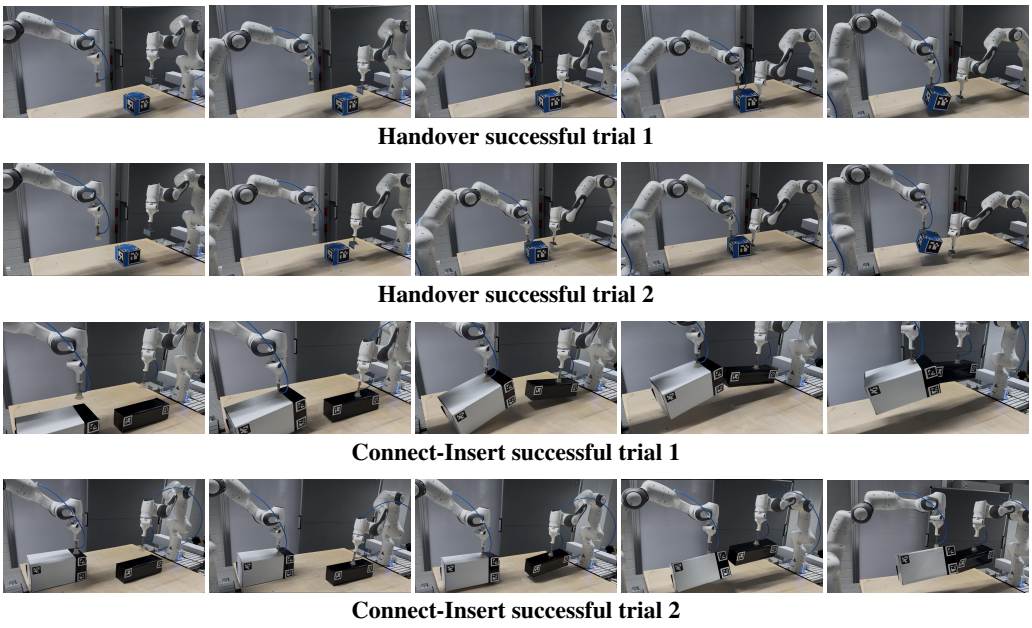


Figure 10: **Real-world robot inference based on our framework.**

To reduce such discrepancies, we use a fixed end-effector orientation, keeping the gripper as close as possible to a downward-facing direction during motion. This additional constraint helps limit the kinematic solutions and makes the real-world behavior more consistent with the simulated behavior.

In addition, we present several representative segments for handover and insert tasks, as shown in Figure 10, allowing clear visualization of the real-world robotic execution process. During real-world assembly, the box may be lifted or shaken, leading to temporary distance-based success even when full insertion is not achieved. Therefore, we refer to the real-robot implementation of `insert` as **Connect-Insert**, which reflects this practical phenomenon and is also considered a successful execution in our evaluation.

A.2 Environment API

We adopt a Gymnasium-style environment interface (Towers et al., 2024). The MuJoCo Point tasks follow implementation conventions inspired by OGBench, while the tasks themselves are newly designed; all remaining tasks and associated code in MuJoCo are implemented independently. For the PyBullet component, Panda-gym (Gallouédec et al., 2021) serves only as a reference for code architecture and interface organization as a single-agent online platform; we develop the new task designs and their multi-agent extensions. A minimal continuous rollout is:

```
from mpg_core import make_env

env = make_env("MuJoCoPointMultiGoalArena-v1")
obs, info = env.reset(seed=42)
action = env.action_space.sample()
next_obs, reward, terminated, truncated, info = env.step(action)
env.close()
```

A.3 Role-Source Assignment

Exchanging the sources assigned to two agents can change the outcome substantially, so role-source assignment is not a symmetric operation. We therefore examine it on the 14 source-study tasks of Section 5.2 (Figure 11), averaging Panel A over the seven methods to reduce the influence of any single strong method. In the dual-arm PyBullet Insert and Connect tasks, the two arms contribute roughly symmetrically. In Handover, where arm 1 first pushes the object into the workspace of

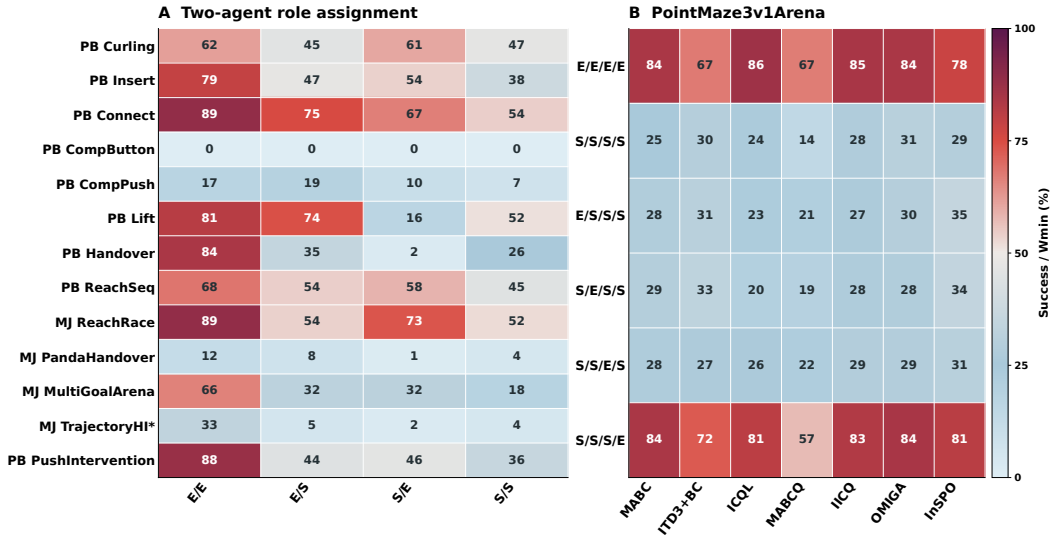


Figure 11: **Role-source assignment across 14 tasks.** (A) Thirteen two-agent tasks, averaged over the seven methods: success rate, or weaker-side win rate $W_{\min}$ for *Competition* tasks; *completion count $\times 100$. (B) Six four-role configurations of PointMaze3v1Arena, $W_{\min}$ per method. E and S denote PretrainedExpert and PretrainedSuboptimal. All pools hold 50k transitions; cells are five-seed means.

arm 2 and arm 2 then acquires it, the asymmetry is pronounced: method-averaged success is 35% when the pushing arm receives the expert source (E/S) but 2% under the reverse assignment (S/E). This explicit division of roles corresponds to a clearer asymmetry between the agents, suggesting that functional specialization within a task can be associated with asymmetric contributions. In PointMaze3v1Arena (Panel B), upgrading only the fourth seat, the single-agent side, to expert data recovers nearly the all-expert level (S/S/S/E versus E/E/E/E), whereas upgrading any one of the other three seats leaves $W_{\min}$ close to the all-suboptimal level.

A.4 Definitions and Estimation of Structural Diagnostics

Reward-event alignment. Let b_i be agent i's median per-step reward over the complete episodes $\mathcal{E}$ of a pool. For episode e of length T_e, a non-baseline return and its pairwise balance are

$$\widetilde{u}_i^e = \sum_{t=1}^{T_e} \big(r_{e,t}^i - b_i\big), \qquad u_i^e = \widetilde{u}_i^e \, \mathbf{1}\big[\big|\widetilde{u}_i^e\big| > 10^{-5} \max\big(1, |b_i|\big)\big] \,,$$
$$q_{ij}^e = \frac{\min\big(|u_i^e|, |u_j^e|\big)}{\max\big(|u_i^e|, |u_j^e|\big)}, \tag{4}$$

with $q_{ij}^e = 0$ when the denominator vanishes. Over $\mathcal{P}_R = \{(i,j) : i < j\}$,

$$R^{\pm} = \frac{1}{|\mathcal{E}|\,|\mathcal{P}_R|} \sum_{e \in \mathcal{E}} \sum_{(i,j) \in \mathcal{P}_R} q_{ij}^e \, \mathbf{1}\big[\pm u_i^e u_j^e > 0\big]. \tag{5}$$

Zero-event episodes enter the mean, so both components vanish in a pool without non-baseline events; budget-truncated episodes are excluded.

Conditional action dependence. Two predictors estimate a_i from the current central state, local observation and goals; the augmented one additionally receives a_j. With L_i^{base} and L_{ij}^{aug} the mean losses on the same held-out transitions and $\mathcal{P}_A$ the valid directed pairs,

$$g_{ij} = 1 - \frac{L_{ij}^{\text{aug}}}{L_i^{\text{base}}}, \qquad C_A = \max\left(0, \; \frac{1}{|\mathcal{P}_A|} \sum_{(i,j) \in \mathcal{P}_A} g_{ij}\right), \tag{6}$$

using squared loss for continuous actions and categorical log loss for discrete ones. A pair is invalid if its training target is constant, a loss is nonfinite, or $L_i^{\text{base}} \leq 10^{-10}$; negative gains are retained, and C_A is missing when no pair is valid.

Phase concentration. Held-out time $t/\max(T_e - 1, 1)$, with t zero-indexed, is partitioned into five bins. Reusing the fitted predictors, $g_{ij,b}$ is computed from losses restricted to bin b, and $\mathcal{B}$ collects the bins holding at least 16 held-out transitions and one valid pair, with $B = |\mathcal{B}|$. Then

$$d_b = \max\left(0,\ \frac{1}{|\mathcal{P}_b|}\sum_{(i,j)\in\mathcal{P}_b} g_{ij,b}\right), \qquad p_b = \frac{d_b}{\sum_{k\in\mathcal{B}} d_k},$$
$$C_P = \left(\frac{1}{B}\sum_{b\in\mathcal{B}} d_b\right)\left(1 - \frac{H(p)}{\log B}\right), \qquad H(p) = -\sum_{b:\, p_b>0} p_b \log p_b. \tag{7}$$

C_P is set to zero when $\sum_{b\in\mathcal{B}} d_b \leq 10^{-10}$ and reported missing when $B < 2$.

A.5 Taskwise Responses to Data Composition

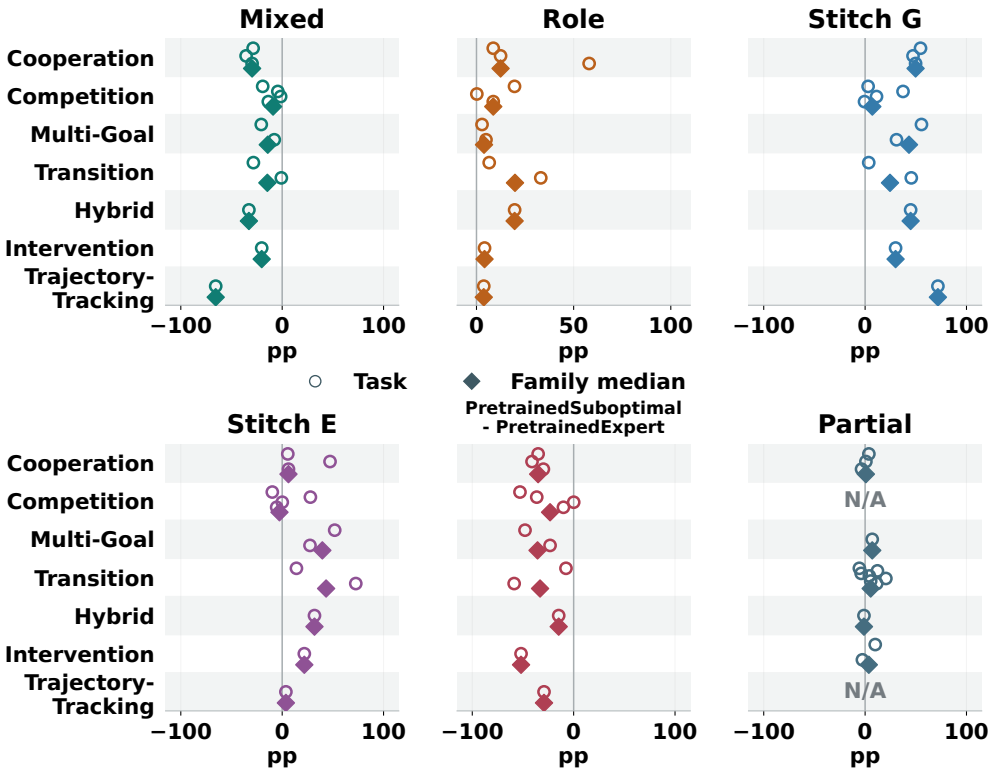


Figure 12: **Task-matched responses to data construction and observation.** The six panels show *Mixed*−*Guided*, the absolute role-swap response $|E/S - S/E|$, Stitch $(R \to G) - (G \to R)$, Stitch $(R \to E) - (E \to R)$, *PretrainedSuboptimal*−*PretrainedExpert*, and Relative−Partial. Open circles are task responses averaged over seven methods; diamonds are within-family medians. All panels use five seeds, 50k transitions, and 50k updates. The first five panels cover the 14 continuous source-study tasks (13 for the two-agent role swap), and the last covers 14 of the 16 observation-comparison tasks. N/A marks families without a paired task, not a zero response.

In Figure 12, *Mixed* falls below *Guided* on every source-study task, and *PretrainedSuboptimal* falls below *PretrainedExpert* on all but one task, where the response is near zero. An equal transition

budget therefore does not make two collection procedures interchangeable, and the six contrasts respond differently. Among families with more than one source-study task, role-swap sensitivity is highest in *Transition* and lowest in *Multi-Goal*, with *Cooperation* and *Competition* in between. Placing the purposeful controller in the later phase outperforms the reverse Stitch order on nearly every task; the few reversals occur only in *Competition*. RELATIVE exceeds PARTIAL on most tasks of the observation cohort, again most consistently in *Transition*.

A.6 COMPLETE OFFLINE RESULTS

Tables 3–5 give the per-task results that Figure 4 aggregates by family. Good, Medium, and Poor follow the rule of Section 4.1, and each row's pool code names the source actually used. Pools hold 50k transitions, or 500k for MuJoCo Panda and Point, while training is 50k updates throughout. Within a task-quality row, all seven methods share the pool, the five seeds, and the held-out protocol.

Table 3: **PyBullet taskwise results.** Five-seed Final mean with sample standard deviation for Good, Medium, and Poor. The code after each quality letter names the pool actually used: E/S/G/M/R for PretrainedExpert, PretrainedSuboptimal, Guided, Mixed, and Random. Bold marks the largest mean in a row where the methods differ, and is not a significance test. Rates carry one decimal, repeated-pattern counts two.

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Asyn dual *(Success%)*							
G / E	85.2 ±4.8	93.8 ±4.1	**98.8** ±1.1	63.6 ±14.1	86.4 ±5.0	93.6 ±2.5	98.6 ±1.7
M / S	63.0 ±3.7	**75.0** ±8.4	56.8 ±8.9	8.8 ±4.8	61.6 ±7.4	60.0 ±4.8	60.6 ±9.3
P / R	*all methods* 0.0 ± 0.0						
CompetitionButton dual *($W_{\min}$%)*							
G / E	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**0.4** ±0.9
M / S	**0.2** ±0.4	**0.2** ±0.4	0.0 ±0.0	0.0 ±0.0	**0.2** ±0.4	0.0 ±0.0	**0.2** ±0.4
P / R	0.0 ±0.0	**0.4** ±0.9	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.2 ±0.4
CompetitionPush dual *($W_{\min}$%)*							
G / E	14.0 ±3.4	**32.2** ±28.6	17.0 ±5.8	8.4 ±4.5	14.2 ±2.3	13.4 ±2.1	23.2 ±3.8
M / S	7.4 ±2.9	7.6 ±6.6	4.4 ±1.8	2.8 ±2.2	4.8 ±2.8	4.4 ±1.9	**20.4** ±5.7
P / R	0.0 ±0.0	11.4 ±11.2	**27.6** ±8.6	0.4 ±0.9	0.0 ±0.0	0.0 ±0.0	0.6 ±1.3
Connect dual *(Success%)*							
G / E	89.6 ±3.2	**93.6** ±2.9	85.4 ±9.2	80.6 ±3.7	91.8 ±2.6	91.4 ±3.1	79.0 ±20.1
M / S	48.0 ±6.0	50.8 ±26.3	52.8 ±3.1	48.6 ±2.7	50.8 ±3.6	49.2 ±3.1	**77.8** ±5.3
P / R	*all methods* 0.0 ± 0.0						
Curling dual *(Success%)*							
G / E	84.4 ±2.7	83.2 ±11.5	50.6 ±19.1	43.6 ±13.2	82.0 ±7.5	**93.4** ±4.6	16.6 ±15.8
M / S	29.4 ±2.7	**75.0** ±9.0	52.4 ±7.7	27.6 ±6.8	29.2 ±7.9	43.2 ±11.0	62.6 ±15.4
P / R	0.0 ±0.0	1.2 ±1.3	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**7.4** ±6.3
GOODLettersRepeat quad *(Patterns/episode)*							
G / G	70.97 ±0.07	73.34 ±1.41	70.16 ±1.08	77.02 ±0.75	71.19 ±0.42	71.00 ±0.00	**80.23** ±1.72
M / M	24.32 ±7.52	0.00 ±0.00	**73.58** ±2.17	0.00 ±0.00	21.46 ±11.02	71.48 ±0.58	0.00 ±0.00
P / R	*all methods* 0.00 ± 0.00						
HILetters dual hiMulti *(Success%)*							
G / G	**100.0** ±0.0	0.0 ±0.0	98.8 ±1.8	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	0.0 ±0.0
M / M	44.6 ±17.8	35.8 ±33.6	16.4 ±33.9	0.0 ±0.0	81.8 ±11.5	**100.0** ±0.0	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
HILettersRepeat dual hiMulti *(Patterns/episode)*							
G / E	76.08 ±1.39	80.24 ±2.08	81.70 ±0.19	74.14 ±7.60	75.59 ±1.46	80.61 ±0.50	**81.81** ±0.20
M / M	48.92 ±5.39	0.05 ±0.05	69.50 ±1.96	0.17 ±0.38	49.56 ±7.77	**70.17** ±1.85	15.23 ±33.96
P / R	*all methods* 0.00 ± 0.00						

continued on next page

Table 3 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Handover dual *(Success%)*							
G / E	99.2 ±0.8	**99.8** ±0.4	77.6 ±3.6	42.2 ±18.7	99.0 ±0.7	98.8 ±1.1	82.8 ±17.5
M / S	35.8 ±4.5	**38.0** ±5.1	3.8 ±2.2	0.0 ±0.0	32.8 ±4.1	33.6 ±3.4	35.4 ±4.6
P / R	*all methods* 0.0 ± 0.0						
Handover triple *(Success%)*							
G / G	**91.2** ±4.3	10.2 ±11.1	0.0 ±0.0	0.0 ±0.0	91.0 ±4.5	88.4 ±6.7	41.2 ±10.3
M / M	39.8 ±11.3	18.4 ±4.2	0.0 ±0.0	0.0 ±0.0	38.4 ±16.3	**65.2** ±12.7	3.2 ±3.6
P / R	*all methods* 0.0 ± 0.0						
HandoverActiveSuction dual *(Success%)*							
G / G	81.2 ±11.0	2.0 ±3.4	0.4 ±0.9	33.0 ±18.6	92.8 ±8.3	**98.2** ±2.0	53.6 ±48.5
M / M	58.4 ±13.1	23.4 ±21.4	0.0 ±0.0	5.8 ±4.9	48.4 ±33.0	**76.4** ±25.1	32.6 ±27.4
P / R	*all methods* 0.0 ± 0.0						
HandoverIntervention dual *(Success%)*							
G / G	99.6 ±0.5	2.4 ±3.4	0.0 ±0.0	35.6 ±7.7	99.6 ±0.9	**100.0** ±0.0	95.2 ±4.4
M / M	57.6 ±7.4	52.2 ±10.2	0.0 ±0.0	2.8 ±6.3	57.4 ±7.8	**93.8** ±7.7	22.8 ±20.1
P / R	*all methods* 0.0 ± 0.0						
HandoverPush dual *(Success%)*							
G / E	92.4 ±0.9	44.6 ±41.9	72.0 ±7.3	26.2 ±7.9	91.0 ±4.5	**93.4** ±1.3	62.0 ±18.7
M / M	39.4 ±9.2	21.6 ±2.1	0.0 ±0.0	0.0 ±0.0	32.8 ±13.8	**95.4** ±3.9	13.4 ±24.5
P / R	*all methods* 0.0 ± 0.0						
HandoverPush quad *(Success%)*							
G / G	99.4 ±1.3	0.6 ±1.3	0.0 ±0.0	19.4 ±12.1	**100.0** ±0.0	99.2 ±0.8	1.6 ±2.6
M / M	31.0 ±12.0	12.6 ±10.3	0.0 ±0.0	0.0 ±0.0	23.0 ±3.9	**97.2** ±1.1	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
HandoverPush triple *(Success%)*							
G / E	79.6 ±2.3	73.6 ±10.2	68.2 ±8.9	1.6 ±1.1	78.6 ±4.6	**88.4** ±3.1	67.2 ±12.9
M / S	32.4 ±3.6	0.4 ±0.9	**56.2** ±6.9	2.6 ±4.8	33.8 ±6.3	37.6 ±6.5	4.0 ±7.8
P / R	*all methods* 0.0 ± 0.0						
Insert dual *(Success%)*							
G / E	**92.8** ±4.3	45.4 ±24.7	82.2 ±2.8	62.8 ±6.5	92.4 ±2.9	90.8 ±5.2	91.4 ±5.9
M / S	35.4 ±6.7	30.2 ±4.3	44.2 ±3.6	17.0 ±6.0	31.2 ±9.3	44.2 ±5.5	**61.6** ±12.1
P / R	*all methods* 0.0 ± 0.0						
InsertActiveSuction dual *(Success%)*							
G / E	91.4 ±1.7	0.4 ±0.5	64.0 ±6.4	4.0 ±3.4	90.0 ±2.4	91.2 ±2.9	**93.2** ±3.2
M / S	30.8 ±5.4	4.4 ±1.9	21.2 ±3.3	2.4 ±1.5	32.6 ±5.6	29.8 ±4.5	**36.8** ±7.0
P / R	*all methods* 0.0 ± 0.0						
InsertIntervention dual *(Success%)*							
G / E	85.6 ±4.6	66.8 ±7.2	78.6 ±8.3	62.4 ±2.5	89.0 ±3.2	86.2 ±2.8	**93.8** ±3.3
M / S	44.6 ±4.8	11.6 ±8.5	34.0 ±19.5	48.6 ±4.1	42.0 ±4.7	41.4 ±5.5	**54.4** ±5.3
P / R	*all methods* 0.0 ± 0.0						
Lift dual *(Success%)*							
G / E	81.6 ±3.9	85.6 ±19.4	86.0 ±3.1	39.4 ±6.0	81.8 ±5.8	84.8 ±4.0	**97.0** ±1.2
M / S	51.4 ±6.5	**86.8** ±3.8	46.4 ±10.0	7.4 ±7.0	52.4 ±6.1	52.4 ±6.8	72.8 ±8.3
P / R	0.0 ±0.0	0.0 ±0.0	0.4 ±0.5	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**1.8** ±2.5
Lift quad *(Success%)*							
G / E	99.6 ±0.5	99.6 ±0.5	99.0 ±0.7	98.6 ±0.5	99.6 ±0.9	99.4 ±0.9	**100.0** ±0.0
M / S	30.6 ±5.8	**78.8** ±5.0	51.6 ±4.7	39.2 ±7.3	29.6 ±3.8	36.4 ±5.6	61.6 ±7.2
P / R	*all methods* 0.0 ± 0.0						

continued on next page

Table 3 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Lift triple *(Success%)*							
G / E	99.8 ±0.4	**100.0** ±0.0	99.8 ±0.4	99.0 ±1.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
M / S	26.0 ±4.4	**70.2** ±7.4	40.4 ±3.2	55.2 ±11.3	26.6 ±4.0	40.0 ±5.8	61.4 ±4.5
P / R	0.4 ±0.5	9.8 ±1.6	23.2 ±6.9	0.0 ±0.0	0.2 ±0.4	0.0 ±0.0	**37.2** ±16.0
LiftActiveSuction dual *(Success%)*							
G / G	98.8 ±2.2	0.0 ±0.0	76.2 ±24.4	84.6 ±6.8	99.2 ±1.1	**99.4** ±1.3	99.0 ±2.2
M / M	27.0 ±8.6	2.0 ±1.2	23.2 ±19.7	5.2 ±4.0	16.4 ±3.6	**97.8** ±2.9	90.6 ±3.4
P / R	*all methods* 0.0 ± 0.0						
LiftIntervention dual *(Success%)*							
G / G	**100.0** ±0.0	0.0 ±0.0	88.4 ±9.2	73.0 ±3.7	99.8 ±0.4	96.0 ±3.4	98.8 ±1.1
M / M	12.6 ±8.1	24.0 ±9.2	30.6 ±6.2	0.8 ±1.3	12.6 ±4.3	**98.4** ±3.6	96.6 ±1.1
P / R	*all methods* 0.0 ± 0.0						
Push dual *(Success%)*							
G / E	97.8 ±0.8	**98.6** ±0.9	90.6 ±6.7	55.0 ±4.2	97.6 ±1.7	98.0 ±1.6	89.0 ±9.1
M / M	33.2 ±5.8	76.2 ±16.4	77.4 ±4.0	44.2 ±13.0	38.4 ±5.3	**96.4** ±2.3	76.6 ±29.8
P / R	0.6 ±0.9	3.2 ±2.2	8.8 ±4.6	1.6 ±3.0	0.2 ±0.4	0.0 ±0.0	**15.4** ±10.2
PushButton dual *(Success%)*							
G / E	**100.0** ±0.0	5.8 ±7.8	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	99.8 ±0.4
M / S	37.0 ±5.1	0.2 ±0.4	31.6 ±5.8	**39.6** ±5.4	36.0 ±5.2	38.4 ±4.5	24.8 ±8.0
P / R	0.0 ±0.0	1.2 ±2.7	**3.2** ±4.4	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
PushIntervention dual *(Success%)*							
G / E	91.4 ±3.4	**95.6** ±3.4	86.4 ±4.5	63.2 ±4.7	91.6 ±2.3	91.8 ±1.5	93.2 ±2.3
M / S	29.4 ±3.0	**43.4** ±4.8	39.8 ±6.7	36.4 ±6.1	29.4 ±2.8	29.6 ±3.0	42.0 ±4.4
P / R	0.6 ±0.5	3.0 ±3.1	7.0 ±4.0	0.8 ±1.3	0.6 ±0.9	0.4 ±0.5	**14.2** ±16.6
PushSequential dual *(Success%)*							
G / E	94.6 ±1.5	19.0 ±22.6	95.8 ±2.3	95.2 ±3.3	89.4 ±3.0	97.6 ±1.8	**98.6** ±2.1
M / M	2.2 ±1.5	15.8 ±11.9	36.6 ±39.8	0.0 ±0.0	0.4 ±0.9	**87.0** ±6.8	41.2 ±25.3
P / R	0.0 ±0.0	**0.2** ±0.4	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Reach dual *(Success%)*							
G / E	**98.4** ±1.3	9.6 ±5.6	98.2 ±1.6	98.2 ±1.1	**98.4** ±1.3	**98.4** ±1.3	**98.4** ±1.3
M / M	88.8 ±2.2	65.2 ±10.1	**98.0** ±1.0	97.6 ±1.3	90.6 ±4.7	**98.0** ±1.0	**98.0** ±1.0
P / R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.2 ±0.4	**1.2** ±0.8
ReachAvoid dual *(Success%)*							
G / E	**100.0** ±0.0	92.0 ±6.8	92.0 ±6.1	99.6 ±0.5	99.8 ±0.4	**100.0** ±0.0	99.6 ±0.9
M / M	48.4 ±12.1	4.4 ±9.8	11.2 ±15.8	19.8 ±11.5	41.4 ±13.2	**85.2** ±2.6	0.2 ±0.4
P / R	*all methods* 0.0 ± 0.0						
ReachSequential dual *(Success%)*							
G / E	73.0 ±6.3	36.2 ±27.7	**83.2** ±2.9	73.6 ±2.3	71.6 ±5.4	77.6 ±4.5	76.0 ±3.9
M / S	33.2 ±2.0	**66.2** ±7.9	65.4 ±6.7	32.4 ±4.5	33.8 ±3.0	38.2 ±3.7	56.6 ±4.3
P / R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**0.4** ±0.9
Soccer dual *($W_{\min}$%)*							
G / G	23.8 ±4.6	3.6 ±2.1	5.6 ±2.1	9.0 ±2.9	**24.0** ±3.1	10.2 ±3.3	20.2 ±6.8
M / S	27.6 ±3.0	14.0 ±5.3	**31.6** ±5.1	9.8 ±8.8	27.6 ±3.4	28.0 ±3.4	27.2 ±1.9
P / R	1.0 ±0.7	**20.0** ±5.6	0.4 ±0.5	1.2 ±1.3	1.0 ±0.7	0.4 ±0.5	13.6 ±5.7
Stack dual *(Success%)*							
G / E	99.4 ±0.5	96.8 ±2.0	83.0 ±4.9	69.6 ±6.0	98.6 ±1.1	**99.6** ±0.5	96.6 ±1.1
M / S	40.4 ±7.1	**82.6** ±5.5	37.6 ±7.2	31.8 ±4.8	40.2 ±9.8	44.2 ±4.7	72.2 ±6.9
P / R	*all methods* 0.0 ± 0.0						

continued on next page

Table 3 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
TicTacToe dual *($W_{\min}$%)*							
G / G	48.4 ±15.3	**71.6** ±18.6	34.0 ±13.4	48.0 ±4.8	43.2 ±6.5	21.8 ±10.4	65.6 ±29.9
M / M	3.2 ±2.7	21.8 ±11.2	5.0 ±8.1	**34.6** ±15.6	2.2 ±1.8	3.6 ±3.1	0.0 ±0.0
P / R	0.0 ±0.0	**13.4** ±10.9	0.0 ±0.0	0.4 ±0.5	0.0 ±0.0	0.0 ±0.0	3.8 ±4.7
TrajectoryMPG triple *(Patterns/episode)*							
G / G	27.76 ±0.24	0.04 ±0.04	27.67 ±0.31	**28.96** ±0.77	27.67 ±0.29	27.64 ±0.15	0.00 ±0.00
M / M	18.46 ±2.53	0.00 ±0.00	27.06 ±0.60	0.00 ±0.00	16.65 ±5.31	**27.50** ±0.28	0.00 ±0.00
P / R	*all methods* 0.00 ± 0.00						
WindowsMultiGoal dual *(Success%)*							
G / E	57.8 ±27.0	11.6 ±18.0	**99.8** ±0.4	98.6 ±2.6	70.2 ±12.4	95.8 ±7.2	58.6 ±53.5
M / M	27.4 ±7.6	3.2 ±6.1	0.0 ±0.0	24.0 ±16.6	21.0 ±6.6	**100.0** ±0.0	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						

Table 4: **MuJoCo taskwise results.** Panda and Point pools hold 500k transitions and the two DualGripper proxies 50k, while training is 50k updates throughout. Conventions follow Table 3.

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
DualGripperHandoverProxy *(Success%)*							
G / E	91.0 ±11.6	31.0 ±37.1	35.6 ±38.5	97.0 ±6.2	99.6 ±0.9	**100.0** ±0.0	2.0 ±3.1
M / M	74.2 ±26.1	30.6 ±43.7	0.0 ±0.0	99.6 ±0.9	66.6 ±21.5	**100.0** ±0.0	80.0 ±44.7
P / R	*all methods* 0.0 ± 0.0						
DualGripperValveTurnCloseProxy *(Success%)*							
G / E	**100.0** ±0.0	60.0 ±54.8	43.8 ±49.7	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	60.0 ±54.8
M / M	15.6 ±24.4	3.8 ±4.5	19.4 ±43.4	13.2 ±12.6	25.2 ±22.8	**100.0** ±0.0	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaCableArc *(Success%)*							
G / G	**58.6** ±44.7	0.0 ±0.0	0.0 ±0.0	0.2 ±0.4	47.6 ±33.7	7.4 ±16.0	1.6 ±3.6
M / S	0.2 ±0.4	0.2 ±0.4	0.4 ±0.9	0.0 ±0.0	**2.0** ±2.5	0.0 ±0.0	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaCablePlace *(Success%)*							
G / G	2.0 ±3.9	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	2.4 ±5.4	**7.2** ±15.0	0.0 ±0.0
M / S	3.2 ±3.6	0.6 ±1.3	0.0 ±0.0	0.2 ±0.4	**7.2** ±8.8	0.8 ±1.8	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaCableSShape *(Success%)*							
G / G	36.0 ±49.8	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	33.2 ±46.7	**80.0** ±44.7	0.0 ±0.0
M / S	0.6 ±0.9	**3.2** ±6.6	1.4 ±2.6	0.0 ±0.0	1.6 ±1.8	0.2 ±0.4	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaCableUShape *(Success%)*							
G / G	31.6 ±20.2	0.0 ±0.0	0.2 ±0.4	2.8 ±2.8	56.0 ±16.2	**65.4** ±22.5	0.0 ±0.0
M / M	8.2 ±6.7	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	11.2 ±15.3	**63.2** ±31.6	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaDualLift *(Success%)*							
G / G	23.2 ±38.2	24.6 ±9.3	5.4 ±12.1	**93.8** ±3.5	6.6 ±13.7	21.8 ±37.8	3.4 ±4.4
M / S	2.0 ±1.6	**20.4** ±43.4	1.0 ±2.2	2.4 ±4.8	14.4 ±31.6	4.0 ±6.4	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaHandover *(Success%)*							
G / E	32.6 ±46.2	2.0 ±2.0	0.0 ±0.0	**65.8** ±35.2	31.4 ±45.0	22.4 ±23.4	0.0 ±0.0
M / S	2.2 ±4.4	32.8 ±43.8	0.0 ±0.0	**53.0** ±48.7	1.4 ±3.1	1.4 ±3.1	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						

continued on next page

Table 4 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
PandaSlideHandover *(Success%)*							
G / G	**100.0** ±0.0	0.0 ±0.0	0.0 ±0.0	1.8 ±4.0	**100.0** ±0.0	**100.0** ±0.0	6.4 ±10.1
M / M	94.8 ±10.0	37.0 ±33.9	0.0 ±0.0	0.0 ±0.0	94.2 ±8.6	**100.0** ±0.0	0.0 ±0.0
P / R	*all methods* 0.0 ± 0.0						
PandaStack *(Success%)*							
G / G	80.0 ±43.6	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**100.0** ±0.0	75.4 ±15.9	0.2 ±0.4
M / S	*all methods* 0.0 ± 0.0						
P / R	*all methods* 0.0 ± 0.0						
PandaValveClose *(Success%)*							
G / E	74.8 ±35.7	88.6 ±15.9	18.4 ±41.1	**100.0** ±0.0	82.8 ±24.1	49.6 ±42.4	0.0 ±0.0
M / S	51.8 ±31.1	**85.2** ±33.1	26.4 ±37.1	69.6 ±41.6	56.8 ±27.0	61.8 ±23.7	0.4 ±0.5
P / R	*all methods* 0.0 ± 0.0						
PointHeterogeneousMultiGoal *(Success%)*							
G / G	**98.8** ±2.2	97.0 ±2.5	4.4 ±1.9	4.4 ±2.6	94.4 ±5.2	82.6 ±9.0	78.4 ±5.4
M / S	22.8 ±4.7	**33.2** ±2.9	2.4 ±1.5	3.6 ±1.5	22.2 ±6.0	31.0 ±6.6	28.0 ±4.3
P / R	*all methods* 0.0 ± 0.0						
PointMaze3v1Arena *($W_{\min}$%)*							
G / E	83.6 ±1.5	**90.0** ±1.0	87.4 ±3.4	89.4 ±2.2	83.4 ±2.5	85.2 ±2.3	84.8 ±3.6
M / S	26.0 ±4.8	**57.4** ±5.2	19.6 ±3.4	52.8 ±5.7	28.8 ±3.7	28.0 ±7.0	52.6 ±8.3
P / R	0.0 ±0.0	45.8 ±8.8	11.2 ±5.5	59.2 ±9.7	0.2 ±0.4	0.2 ±0.4	**64.8** ±11.8
PointMaze3v1Medium *($W_{\min}$%)*							
G / G	**92.2** ±4.0	89.4 ±4.3	61.6 ±5.6	61.2 ±1.5	90.4 ±4.7	89.0 ±6.0	88.0 ±3.9
M / S	**61.2** ±7.7	59.2 ±4.8	54.4 ±1.9	40.0 ±5.8	61.0 ±7.3	60.0 ±6.9	52.2 ±4.4
P / R	0.2 ±0.4	26.0 ±6.6	8.2 ±2.9	**34.4** ±5.3	0.2 ±0.4	0.4 ±0.5	24.6 ±8.2
PointMultiGoalArena *(Success%)*							
G / E	95.0 ±1.4	**96.4** ±1.9	86.8 ±3.5	52.0 ±6.9	95.2 ±1.5	94.6 ±4.0	90.0 ±2.0
M / S	30.2 ±5.0	**55.2** ±3.7	41.0 ±6.8	39.4 ±10.2	32.0 ±5.0	29.6 ±4.9	51.6 ±5.0
P / R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	**3.8** ±1.6	0.0 ±0.0	0.0 ±0.0	0.6 ±0.5
PointMultiGoalOrder *(Success%)*							
G / E	80.0 ±5.0	**88.8** ±5.1	79.6 ±2.1	80.6 ±1.9	81.0 ±6.3	75.4 ±5.5	88.2 ±3.9
M / S	49.8 ±2.6	55.4 ±3.5	48.6 ±6.1	59.2 ±6.9	49.6 ±1.1	54.0 ±2.4	**62.6** ±3.5
P / R	*all methods* 0.0 ± 0.0						
PointMultiGoalParallel *(Success%)*							
G / E	76.8 ±5.1	82.0 ±4.1	82.6 ±5.0	**90.6** ±1.7	77.6 ±3.6	75.8 ±4.0	84.6 ±4.4
M / S	39.2 ±4.7	41.8 ±3.7	37.8 ±4.1	**56.6** ±3.9	38.6 ±3.8	40.6 ±3.6	51.0 ±8.3
P / R	*all methods* 0.0 ± 0.0						
PointPushMultiGoal *(Success%)*							
G / E	30.2 ±4.1	**45.8** ±9.1	0.0 ±0.0	0.0 ±0.0	26.6 ±2.5	43.0 ±9.2	4.8 ±2.6
M / M	38.0 ±7.0	**43.6** ±12.0	0.0 ±0.0	0.0 ±0.0	34.0 ±8.5	0.0 ±0.0	0.4 ±0.9
P / R	*all methods* 0.0 ± 0.0						
PointPushRace *($W_{\min}$%)*							
G / G	87.4 ±6.8	86.6 ±0.9	4.4 ±3.2	1.4 ±1.1	**90.8** ±3.8	84.2 ±6.4	65.4 ±10.3
M / S	22.2 ±3.7	**43.8** ±3.1	4.8 ±0.8	3.0 ±0.7	22.6 ±3.9	31.2 ±5.6	24.4 ±1.9
P / R	*all methods* 0.0 ± 0.0						
PointReachAvoidRace *($W_{\min}$%)*							
G / E	**94.2** ±3.2	87.4 ±2.9	94.0 ±2.5	77.4 ±3.0	93.6 ±3.8	94.0 ±3.5	92.8 ±2.7
M / M	92.6 ±3.0	70.6 ±4.9	**94.2** ±2.0	57.0 ±7.2	93.2 ±2.2	93.2 ±2.4	92.4 ±2.4
P / R	0.2 ±0.4	25.6 ±6.2	6.4 ±1.5	**74.4** ±13.2	0.0 ±0.0	0.0 ±0.0	22.8 ±8.6

continued on next page

Table 4 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
PointReachRace *($W_{\min}$%)*							
G/E	99.4 ±0.5	91.4 ±2.4	**99.8** ±0.4	87.6 ±3.7	99.2 ±0.8	99.4 ±0.9	98.8 ±0.8
M/S	52.6 ±3.9	**76.4** ±2.3	50.8 ±4.4	66.8 ±6.4	51.6 ±3.8	51.2 ±3.3	64.2 ±3.8
P/R	0.0 ±0.0	23.2 ±7.6	5.2 ±1.8	**84.8** ±2.9	0.0 ±0.0	0.0 ±0.0	21.0 ±11.2
PointSoccerArena *(Success%)*							
G/G	24.4 ±4.5	**27.6** ±5.4	17.2 ±1.8	5.4 ±2.1	25.6 ±3.4	21.6 ±5.1	20.0 ±3.4
M/S	23.0 ±2.4	21.2 ±2.9	25.8 ±5.0	11.4 ±4.4	**27.6** ±2.1	22.2 ±5.0	17.4 ±2.6
P/R	0.0 ±0.0	0.0 ±0.0	0.4 ±0.5	**1.0** ±0.7	0.0 ±0.0	0.0 ±0.0	0.8 ±0.4
PointSoccerIntervention *(Success%)*							
G/G	23.2 ±2.4	28.2 ±5.3	15.8 ±4.4	7.6 ±1.3	27.2 ±5.2	20.0 ±4.8	**34.2** ±2.3
M/S	17.0 ±1.4	16.6 ±2.9	16.4 ±2.6	8.2 ±2.5	**20.2** ±4.3	18.0 ±4.2	12.0 ±2.1
P/R	0.0 ±0.0	0.2 ±0.4	**1.6** ±2.1	0.2 ±0.4	0.0 ±0.0	0.0 ±0.0	1.0 ±0.0
PointTrajectoryHI *(Success%)*							
G/E	75.0 ±6.4	61.4 ±38.3	17.4 ±13.8	0.0 ±0.0	69.2 ±8.5	**97.8** ±4.9	37.2 ±51.0
M/M	74.6 ±16.7	0.0 ±0.0	**100.0** ±0.0	0.0 ±0.0	58.8 ±10.1	91.0 ±4.8	0.0 ±0.0
P/R	*all methods* 0.0 ± 0.0						

Table 5: **Discrete taskwise results.** The six discrete environments use their own three methods. MAICQ appears only where a shared team reward makes its team-value interface well-defined; N/A marks inapplicability and is never counted as zero. Conventions otherwise follow Table 3.

	MABC	MAIQL	MAICQ
CompetitiveButtonRace *($W_{\min}$%)*			
G/E	100.0 ±0.0	100.0 ±0.0	N/A
M/S	35.0 ±6.2	**75.8** ±10.6	N/A
P/R	49.6 ±16.5	**87.4** ±3.6	N/A
KeyedHandoverProtocol *(Success%)*			
G/E	**100.0** ±0.0	**100.0** ±0.0	85.0 ±16.9
M/S	29.6 ±6.7	**82.2** ±5.9	60.4 ±20.3
P/R	*all methods* 0.0 ± 0.0		
KeyedHandoverProtocolCompetitive *($W_{\min}$%)*			
G/E	99.6 ±0.5	**99.8** ±0.4	N/A
M/S	30.6 ±3.0	**32.2** ±4.1	N/A
P/R	0.8 ±0.4	**91.4** ±2.6	N/A
KeyedHandoverProtocolMultiGoal *(Success%)*			
G/E	**100.0** ±0.0	**100.0** ±0.0	85.0 ±10.2
M/S	23.0 ±6.0	**50.6** ±4.6	8.8 ±4.2
P/R	*all methods* 0.0 ± 0.0		
MixedCurling *(Success%)*			
G/E	100.0 ±0.0	100.0 ±0.0	N/A
M/S	49.8 ±7.5	49.8 ±7.5	N/A
P/R	0.0 ±0.0	0.0 ±0.0	N/A
Stack *(Success%)*			
G/E	**100.0** ±0.0	**100.0** ±0.0	97.8 ±4.9
M/S	43.4 ±4.0	**70.4** ±5.2	66.4 ±10.6
P/R	*all methods* 0.0 ± 0.0		

A.7 Goal-Conditioned Results

Tables 6 and 7 report goal-conditioned results on the 50k and 500k collections, and Table 8 gives the per-task values behind Figure 5D.

Table 6: **Goal-conditioned Final success (%) on the 25-task 50k datasets.** Five-seed mean with sample standard deviation; 50k physical transitions and 50k training updates. Each task has two source conditions: E (PretrainedExpert), S (PretrainedSuboptimal), G (Guided), or M (Mixed). Bold marks the better of a task's two sources for that method.

Task	Src	GCBC	GCIQL	GCIVL
MuJoCo				
DualGripperHandoverProxy	M	**98.4** ±2.5	100.0 ±0.0	100.0 ±0.0
	E	71.8 ±31.9	100.0 ±0.0	100.0 ±0.0
DualGripperValveTurnCloseProxy	M	43.6 ±36.9	80.0 ±44.7	100.0 ±0.0
	E	**100.0** ±0.0	**100.0** ±0.0	100.0 ±0.0
PandaCablePlace	G	**25.4** ±9.8	**26.4** ±12.0	**26.0** ±11.5
	M	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
PandaDualLift	G	**63.8** ±10.1	**94.6** ±7.8	**98.6** ±1.5
	M	0.0 ±0.0	40.2 ±25.3	75.2 ±16.6
PandaHandover	S	3.4 ±4.5	6.4 ±11.6	2.8 ±3.3
	E	**10.0** ±7.2	**19.8** ±26.3	**10.6** ±8.8
PandaSlideHandover	G	**0.2** ±0.4	**34.4** ±40.2	**97.8** ±4.4
	M	0.0 ±0.0	1.2 ±1.8	16.6 ±27.8
PandaStack	G	**4.0** ±4.1	**41.4** ±34.0	**81.4** ±20.5
	M	0.0 ±0.0	0.2 ±0.4	0.0 ±0.0
PandaValveClose	S	34.4 ±27.1	**83.0** ±26.9	32.0 ±22.7
	E	**52.0** ±32.8	69.6 ±29.3	**61.8** ±31.0
PointHeterogeneousMultiGoal	G	**80.8** ±11.4	**57.0** ±10.2	**78.8** ±2.8
	M	7.6 ±2.5	5.8 ±2.6	7.6 ±2.2
PointMultiGoalArena	S	30.2 ±6.9	25.8 ±5.4	29.0 ±4.3
	E	**94.0** ±3.2	**92.6** ±2.3	**64.2** ±3.6
PointMultiGoalOrder	S	52.8 ±3.3	49.4 ±4.3	53.6 ±2.1
	E	**78.6** ±4.4	**78.0** ±5.0	**88.2** ±3.0
PointMultiGoalParallel	S	45.0 ±4.5	42.2 ±6.6	54.8 ±6.6
	E	**80.2** ±4.0	**81.2** ±4.8	**81.2** ±4.0
PointPushMultiGoal	G	**81.4** ±1.8	**16.8** ±7.0	**72.6** ±2.6
	M	0.2 ±0.4	0.0 ±0.0	0.0 ±0.0
PyBullet				
Asyn dual	S	58.2 ±5.8	61.6 ±4.7	60.4 ±8.9
	E	**84.6** ±3.4	**94.2** ±2.0	**94.4** ±3.0
Connect dual	G	**100.0** ±0.0	**99.8** ±0.4	**100.0** ±0.0
	M	39.4 ±6.4	67.4 ±12.8	81.4 ±11.2
Handover dual	G	**98.2** ±1.8	**98.2** ±2.4	**100.0** ±0.0
	M	65.4 ±7.4	70.4 ±15.2	79.6 ±13.4
Insert dual	S	30.6 ±7.1	35.4 ±11.3	33.4 ±2.9
	E	**83.2** ±4.4	**87.6** ±8.2	**90.4** ±5.3
Lift dual	S	47.6 ±9.4	64.8 ±5.1	60.6 ±7.7
	E	**85.8** ±2.9	**97.6** ±1.7	**95.4** ±2.3
PushButton dual	S	36.6 ±3.0	38.2 ±4.7	34.0 ±3.3
	E	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
PushSequential dual	G	**98.4** ±1.3	**86.8** ±5.0	**96.2** ±1.6
	M	12.4 ±6.0	66.6 ±6.7	45.0 ±14.4

continued on next page

Table 6 continued

Task	Src	GCBC	GCIQL	GCIVL
Push dual	G	**97.2** ±1.8	**97.8** ±1.9	**98.6** ±0.9
	M	27.4 ±3.6	95.8 ±1.9	96.2 ±3.3
ReachSequential dual	S	34.4 ±3.2	59.8 ±6.1	40.4 ±5.0
	E	**73.6** ±5.5	**86.6** ±2.1	**78.0** ±6.8
Reach dual	G	**98.0** ±1.0	**98.0** ±1.0	**98.0** ±1.0
	M	96.2 ±3.1	97.8 ±0.8	97.0 ±1.2
Stack dual	S	42.4 ±10.7	63.4 ±8.8	54.8 ±7.9
	E	**99.0** ±1.0	**98.6** ±0.9	**99.0** ±1.2
WindowsMultiGoal dual	G	**100.0** ±0.0	100.0 ±0.0	100.0 ±0.0
	M	43.4 ±18.3	100.0 ±0.0	100.0 ±0.0

Table 7: **Goal-conditioned Final success (%) on the 11-task MuJoCo 500k datasets.** Conventions as in Table 6.

Task	Src	GCBC	GCIQL	GCIVL
MuJoCo				
PandaCablePlace	G	**39.6** ±40.7	**76.2** ±28.7	**37.2** ±12.9
	M	0.0 ±0.0	1.2 ±1.8	0.0 ±0.0
PandaDualLift	G	58.0 ±20.3	**99.8** ±0.4	**100.0** ±0.0
	M	**80.2** ±26.2	95.6 ±3.0	98.4 ±1.8
PandaHandover	S	12.6 ±21.7	0.0 ±0.0	6.0 ±7.5
	E	**52.6** ±49.4	**27.8** ±41.8	**26.6** ±32.1
PandaSlideHandover	G	**99.4** ±1.3	**98.8** ±2.7	**100.0** ±0.0
	M	83.2 ±5.5	67.8 ±29.3	90.4 ±14.5
PandaStack	G	**98.8** ±1.8	**98.8** ±1.6	**98.6** ±1.5
	M	71.4 ±17.1	4.8 ±6.1	0.2 ±0.4
PandaValveClose	S	55.4 ±21.3	60.2 ±40.4	44.2 ±22.3
	E	**78.2** ±39.4	**83.4** ±17.4	**64.2** ±40.3
PointHeterogeneousMultiGoal	G	**98.0** ±2.3	**77.4** ±12.5	**94.4** ±4.7
	M	41.6 ±3.7	8.0 ±2.5	54.2 ±8.6
PointMultiGoalArena	S	32.2 ±5.4	59.0 ±2.9	45.6 ±5.8
	E	**96.0** ±2.0	**94.6** ±2.3	**85.8** ±4.8
PointMultiGoalOrder	S	50.2 ±2.3	62.8 ±4.8	69.0 ±2.7
	E	**82.2** ±3.4	**89.4** ±4.0	**94.6** ±2.6
PointMultiGoalParallel	S	40.6 ±5.9	63.8 ±7.0	74.6 ±6.9
	E	**78.2** ±5.5	**88.8** ±4.1	**88.2** ±4.1
PointPushMultiGoal	G	**98.8** ±1.8	**78.4** ±9.5	**82.4** ±4.6
	M	31.6 ±10.5	0.0 ±0.0	18.8 ±7.3

Table 8: **Per-task goal-conditioned results behind Figure 5D.** Twelve data conditions drawn from the eight data constructions, 50k transitions, five-seed Final mean with sample standard deviation. Bold marks the largest mean in a row.

Source	GCBC	GCIQL	GCIVL
PyBullet Insert dual			
PretrainedExpert (E)	81.6 ±4.8	**89.6** ±7.2	89.0 ±3.2
PretrainedSuboptimal (S)	28.4 ±9.4	**28.8** ±4.2	**28.8** ±5.9
Guided (G)	93.8 ±3.1	**98.0** ±0.7	96.0 ±2.2
Mixed (M)	21.4 ±8.0	39.8 ±6.9	**67.8** ±6.8
Random (R)	0.0 ±0.0	0.0 ±0.0	**0.6** ±0.9
SA-Extended	36.8 ±3.6	**43.2** ±5.4	**43.2** ±5.3
Combination E / S	39.8 ±8.5	36.6 ±9.5	**46.0** ±9.8
Combination S / E	37.6 ±6.7	37.8 ±13.2	**48.2** ±14.3
Stitch G→R	0.0 ±0.0	**3.8** ±3.6	2.6 ±2.1
Stitch R→G	40.0 ±3.5	59.6 ±17.4	**80.8** ±4.4
Stitch E→R	1.2 ±1.1	**18.8** ±6.9	15.0 ±5.8
Stitch R→E	52.8 ±9.9	**84.2** ±6.6	84.0 ±5.3
MuJoCo PointMultiGoalArena			
PretrainedExpert (E)	**96.2** ±1.1	92.6 ±3.2	71.8 ±3.1
PretrainedSuboptimal (S)	27.8 ±5.2	24.8 ±4.8	**28.0** ±3.7
Guided (G)	**97.8** ±1.3	96.6 ±1.7	96.4 ±3.4
Mixed (M)	77.4 ±5.4	**86.8** ±2.8	86.4 ±3.2
Random (R)	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
SA-Extended	3.0 ±2.9	**6.0** ±4.8	1.8 ±1.1
Combination E / S	**53.4** ±3.4	52.4 ±5.8	28.4 ±7.4
Combination S / E	**54.2** ±6.4	53.8 ±2.9	33.4 ±3.1
Stitch G→R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Stitch R→G	79.4 ±3.9	96.4 ±2.4	**97.4** ±0.5
Stitch E→R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Stitch R→E	76.4 ±5.7	**94.2** ±2.9	93.0 ±0.7
PyBullet Lift dual			
PretrainedExpert (E)	78.4 ±3.4	**95.0** ±2.2	92.6 ±3.4
PretrainedSuboptimal (S)	49.2 ±4.6	**67.2** ±5.9	54.0 ±9.1
Guided (G)	**99.6** ±0.5	98.8 ±2.2	**99.6** ±0.5
Mixed (M)	69.4 ±8.6	62.6 ±18.3	**94.4** ±3.4
Random (R)	0.0 ±0.0	0.4 ±0.9	**1.0** ±1.0
SA-Extended	25.0 ±5.0	**30.4** ±5.4	27.6 ±3.8
Combination E / S	71.2 ±3.5	**80.6** ±1.8	77.2 ±1.6
Combination S / E	9.4 ±2.7	**34.6** ±4.3	29.0 ±5.4
Stitch G→R	6.2 ±3.7	**39.2** ±6.9	**39.2** ±11.0
Stitch R→G	48.0 ±10.7	60.8 ±14.5	**89.6** ±7.8
Stitch E→R	42.4 ±9.6	**86.8** ±2.6	82.8 ±5.1
Stitch R→E	51.6 ±4.8	**93.4** ±1.5	89.0 ±3.2

continued on next page

Table 8 continued

Source	GCBC	GCIQL	GCIVL
MuJoCo PandaHandover			
PretrainedExpert (E)	12.4 ±15.5	**28.2** ±26.2	3.6 ±3.3
PretrainedSuboptimal (S)	0.8 ±1.3	**13.8** ±12.8	2.0 ±1.6
Guided (G)	78.6 ±33.2	**90.6** ±7.1	81.6 ±15.6
Mixed (M)	0.0 ±0.0	**5.0** ±4.5	1.0 ±0.7
Random (R)	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
SA-Extended	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Combination E / S	2.4 ±2.7	**38.8** ±37.7	20.2 ±9.6
Combination S / E	**7.0** ±9.2	0.6 ±0.9	1.6 ±2.5
Stitch G→R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Stitch R→G	49.2 ±29.1	74.6 ±42.5	**97.6** ±2.2
Stitch E→R	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
Stitch R→E	0.4 ±0.9	14.4 ±16.2	**26.2** ±23.6

A.8 Seed Variability of Aggregate Results

Figures 4 and 5 show sample standard deviations only for the discrete panels. Tables 9–11 report five-seed means and sample standard deviations for the remaining panels; the two-sided 95% confidence interval is the mean ± 1.242 SD.

Table 9: Five-seed means and sample standard deviations behind Figure 4A–C. Each seed contributes one task-equal family mean; parentheses give the number of tasks. Panels A and B are in percent (*Competition* uses $W_{\min}$, one-shot *Trajectory-Tracking* uses episode completion); Panel C counts completed patterns per episode. The two-sided 95% confidence interval over the five seeds is the mean ± 1.242 SD ($t_{0.975,4}/\sqrt{5} = 1.242$). Bold marks the highest mean in each row.

Condition	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Panel A Good data, continuous tasks							
Cooperation (19)	76.6 ±4.8	47.1 ±3.4	53.9 ±6.5	57.3 ±0.7	**77.8** ±4.0	76.8 ±5.2	58.3 ±3.1
Transition (10)	85.2 ±4.7	35.8 ±3.9	35.3 ±3.5	35.1 ±4.0	87.0 ±4.0	**88.2** ±2.5	41.5 ±6.4
Multi-Goal (9)	78.5 ±2.1	53.6 ±2.4	70.2 ±0.7	66.1 ±1.2	78.4 ±1.3	**82.5** ±1.3	75.4 ±6.1
Hybrid (2)	54.4 ±2.5	55.4 ±7.7	33.9 ±9.8	24.5 ±6.5	53.8 ±4.8	**57.5** ±3.4	18.3 ±6.6
Intervention (5)	80.0 ±0.8	38.6 ±3.2	53.8 ±2.0	48.4 ±1.7	81.4 ±1.3	78.8 ±1.7	**83.0** ±1.3
Competition (9)	60.3 ±2.2	**61.4** ±3.7	44.9 ±2.0	42.5 ±1.1	59.9 ±0.5	55.2 ±1.1	59.9 ±3.7
Trajectory-Tracking, one-shot (2)	87.5 ±3.2	30.7 ±19.1	58.1 ±7.6	50.0 ±0.0	84.6 ±4.3	**98.9** ±2.5	18.6 ±25.5
Panel B Medium data, continuous tasks							
Cooperation (19)	28.5 ±3.2	35.0 ±3.1	29.1 ±3.1	23.9 ±0.9	29.9 ±2.6	**47.6** ±1.4	37.4 ±1.9
Transition (10)	47.1 ±4.9	29.0 ±10.2	11.7 ±0.9	17.0 ±4.4	43.3 ±4.9	**66.7** ±3.6	22.9 ±6.3
Multi-Goal (9)	31.1 ±1.8	35.0 ±3.4	29.3 ±6.1	28.3 ±2.5	29.7 ±1.4	**46.5** ±1.2	35.1 ±2.8
Hybrid (2)	26.2 ±1.9	**48.1** ±4.4	39.1 ±3.4	19.5 ±4.7	28.4 ±3.2	32.7 ±3.9	40.0 ±8.7
Intervention (5)	32.2 ±3.1	29.6 ±3.2	24.2 ±5.2	19.4 ±1.3	32.3 ±3.5	**56.2** ±1.6	45.6 ±3.5
Competition (9)	32.6 ±0.7	**39.0** ±1.8	29.4 ±1.0	29.6 ±1.6	32.4 ±1.0	33.3 ±1.3	37.1 ±0.2
Trajectory-Tracking, one-shot (2)	59.6 ±11.2	17.9 ±16.8	58.2 ±17.0	0.0 ±0.0	70.3 ±7.1	**95.5** ±2.4	0.0 ±0.0
Panel C Repeated *Trajectory-Tracking*, 3 tasks *(patterns per episode)*							
Good data	58.27±0.47	51.20±0.92	59.85±0.41	**60.04**±2.51	58.15±0.59	59.75±0.17	54.01±0.58
Medium data	30.57±3.56	0.02±0.02	**56.71**±1.12	0.06±0.13	29.22±4.46	56.38±0.54	5.08±11.32

Table 10: Five-seed means and sample standard deviations behind Figure 5A–B, in percent. Panel A averages ten non-competitive tasks by success rate (the one-shot *Trajectory-Tracking* task by completion); Panel B reports $W_{\min}$ on *Competition* tasks, with the number of tasks in parentheses. The 95% confidence interval is the mean ± 1.242 SD; other conventions follow Table 9.

Source	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Panel A Continuous success, 10 tasks							
PretrainedExpert	76.1 ±1.9	61.4 ±4.9	59.3 ±1.9	42.9 ±4.4	76.8 ±1.6	**83.5** ±2.0	63.8 ±3.5
PretrainedSuboptimal	29.3 ±1.7	40.2 ±6.4	30.1 ±1.5	21.2 ±4.6	30.3 ±1.5	33.3 ±1.8	**42.0** ±1.6
Guided	**96.4** ±3.8	30.7 ±8.4	57.5 ±6.9	69.3 ±2.7	96.0 ±3.0	88.4 ±1.2	76.1 ±2.7
Mixed	33.5 ±1.8	37.8 ±4.7	49.7 ±3.7	23.5 ±2.4	34.2 ±1.7	**80.8** ±2.1	65.6 ±3.5
Random	0.0 ±0.1	0.5 ±0.3	0.5 ±0.3	0.2 ±0.2	0.0 ±0.1	0.0 ±0.1	**2.3** ±1.0
SA-Extended	48.8 ±1.0	12.2 ±1.3	25.8 ±5.5	49.4 ±2.7	49.3 ±1.1	49.6 ±1.3	**52.1** ±1.6
Combo E/S	44.4 ±2.6	44.0 ±6.0	38.8 ±3.5	26.6 ±2.6	43.8 ±2.1	**50.4** ±4.6	46.6 ±3.0
Combo S/E	31.7 ±3.0	39.7 ±2.7	32.6 ±2.5	23.2 ±2.1	31.5 ±2.1	35.3 ±2.1	**42.7** ±2.5
Stitch G→R	3.8 ±0.6	12.4 ±1.0	**32.1** ±1.4	5.1 ±1.0	3.6 ±0.8	15.4 ±2.1	17.2 ±1.6
Stitch R→G	54.7 ±3.3	45.2 ±5.2	51.2 ±6.1	44.4 ±3.1	56.3 ±4.1	**76.5** ±2.7	65.1 ±5.6
Stitch E→R	14.1 ±1.1	20.7 ±3.1	**32.1** ±3.1	9.9 ±1.6	15.0 ±1.3	27.1 ±1.1	24.7 ±2.7
Stitch R→E	51.0 ±1.1	34.3 ±3.2	49.0 ±9.1	15.2 ±4.8	49.9 ±2.8	**76.5** ±5.3	65.1 ±4.4
Panel B *Competition*, $W_{\min}$							
PretrainedExpert (4)	49.2 ±1.3	37.9 ±7.0	**50.5** ±1.8	38.6 ±1.9	49.4 ±0.5	49.2 ±0.6	49.0 ±1.7
PretrainedSuboptimal (4)	21.7 ±1.4	20.4 ±1.4	19.9 ±1.9	17.1 ±2.2	21.2 ±2.0	21.8 ±1.3	**27.8** ±1.7
Guided (4)	52.5 ±2.0	30.6 ±4.6	52.1 ±1.8	51.0 ±0.9	52.4 ±1.1	53.6 ±1.2	**56.5** ±1.0
Mixed (4)	46.5 ±0.5	28.8 ±1.2	38.2 ±1.1	30.9 ±3.1	**46.8** ±0.4	43.0 ±1.1	46.6 ±2.0
Random (4)	4.5 ±1.0	11.7 ±2.4	**13.2** ±2.0	7.5 ±1.1	3.8 ±1.1	4.1 ±0.5	10.7 ±2.0
SA-Extended (4)	39.5 ±0.7	22.9 ±3.9	30.6 ±0.8	35.5 ±1.2	39.2 ±0.5	37.9 ±2.8	**40.4** ±2.1
Combo E/S (3)	22.1 ±1.3	**27.9** ±9.9	26.5 ±2.4	20.9 ±1.8	21.8 ±1.0	22.4 ±1.5	27.1 ±2.0
Combo S/E (3)	28.7 ±1.7	20.3 ±3.4	**31.8** ±1.4	27.4 ±2.1	29.2 ±0.8	27.7 ±2.2	29.5 ±2.0
Stitch G→R (4)	24.3 ±1.9	22.4 ±3.2	**32.9** ±7.1	17.2 ±1.3	24.1 ±2.0	16.9 ±1.4	27.6 ±2.1
Stitch R→G (4)	42.4 ±2.4	26.8 ±2.8	45.0 ±4.1	8.8 ±1.9	40.3 ±3.0	**58.9** ±1.2	32.6 ±4.6
Stitch E→R (4)	27.2 ±1.9	27.8 ±1.8	**37.4** ±2.0	21.4 ±0.8	27.4 ±1.3	28.2 ±1.9	34.1 ±4.7
Stitch R→E (4)	38.2 ±3.8	30.7 ±4.2	34.7 ±3.7	6.9 ±1.1	40.1 ±3.4	**47.0** ±0.7	28.2 ±2.8
3v1 E/S/S/S (1)	28.4 ±6.2	31.0 ±5.3	22.8 ±1.1	20.8 ±5.3	26.8 ±7.6	29.6 ±6.1	**34.8** ±4.0
3v1 S/E/S/S (1)	29.0 ±6.8	33.2 ±2.4	20.0 ±3.8	18.8 ±2.9	27.6 ±7.2	28.2 ±3.3	**34.0** ±5.2
3v1 S/S/E/S (1)	27.6 ±7.3	27.2 ±5.9	26.2 ±3.7	22.2 ±6.3	28.8 ±7.6	28.6 ±4.3	**30.8** ±6.8
3v1 S/S/S/E (1)	84.0 ±1.4	72.4 ±4.2	80.6 ±2.3	57.4 ±3.8	82.6 ±3.4	**84.4** ±0.9	81.0 ±1.7

Table 11: Five-seed means and sample standard deviations behind Figure 5D, in percent: four tasks trained with goal-conditioned relabeling. The 95% confidence interval is the mean ± 1.242 SD; other conventions follow Table 9.

Source	GCBC	GCIQL	GCIVL
PretrainedExpert	67.2 ±2.3	**76.3** ±6.9	64.2 ±1.5
PretrainedSuboptimal	26.6 ±2.5	**33.6** ±3.5	28.2 ±2.2
Guided	92.5 ±8.0	**96.0** ±2.5	93.4 ±3.8
Mixed	42.0 ±4.3	48.5 ±5.7	**62.4** ±1.6
Random	0.0 ±0.0	0.1 ±0.2	**0.4** ±0.1
SA-Extended	16.2 ±1.9	**19.9** ±3.0	18.1 ±1.5
Combo E/S	41.7 ±1.6	**52.1** ±10.5	43.0 ±5.5
Combo S/E	27.1 ±4.1	**31.7** ±3.1	28.1 ±4.1
Stitch G→R	1.6 ±0.9	**10.8** ±2.1	10.4 ±2.3
Stitch R→G	54.1 ±9.3	72.8 ±12.5	**91.3** ±2.7
Stitch E→R	10.9 ±2.3	**26.4** ±1.7	24.4 ±2.0
Stitch R→E	45.3 ±3.7	71.5 ±5.0	**73.0** ±7.3

A.9 WITHIN-OPERATION COMPARISONS OF TASK VARIANTS

Our task design also includes variants of the same base task (Figure 1 and Table 12). *Intervention* adds a bounded, seed-controlled disturbance and applies stricter suction-geometry and workspace checks, while the observation and action spaces stay unchanged. ActiveSuction adds a suction command to each agent's action and changes the attachment and release conditions. We compare each base task with its variants under the same data and training budgets (Tables 13 and 14). The responses depend on both the method and the data pool. Under *Intervention*, 67% of the 55 differences are negative and 27% positive, and 42% are distinguishable from zero. On Lift dual with the Guided pool, ITD3+BC falls from 69.2% to 0.0%, whereas ICQL rises from 57.2% to 88.4%. OMIGA drops markedly on Insert dual, from 93.6% to 68.4% with the Guided pool and from 93.2% to 44.2% with the Mixed pool, but changes little on the other three tasks. Under ActiveSuction, 74% of the 42 differences are negative, and 36% are distinguishable from zero; on Lift dual with the Mixed pool, ITD3+BC falls from 79.2% to 2.0%, whereas OMIGA only falls from 100.0% to 97.8%. To summarize how much the ordering of methods changes, we compute Kendall's τ (Kendall et al., 1938) between the base and variant means of the seven methods. It lies between 0.35 and 0.71 for *Intervention* and between 0.20 and 0.90 for ActiveSuction, so rankings are partly preserved but not fixed.

With the task, simulator, embodiment, and horizon held fixed, the method-dependent responses cannot be explained by platform differences alone, although these comparisons do not separate the rule change from the accompanying change in the collected data.

Table 12: **Interface of the compared tasks.** Task names are abbreviated; the full registered names are listed in Appendix A.24. All entries are dual-arm PyBullet configurations. Actor dimensions include the local goal; action dimensions are per agent.

			Action dim.		
Task	**Horizon**	**Actor dim.**	Base	*Intervention*	ActiveSuction
Lift dual	50	29	3	3	4
Push dual	50	30	3	3	—
Handover dual	80	23	3	3	4
Insert dual	80	50	3	3	4

Table 13: ***Intervention* versus the base task.** Five-seed Final success (%) with sample standard deviation. Base and Variant use pools of the same source class (Guided or Mixed), the same budgets, and the same evaluation protocol. Δ is the within-seed difference Variant $-$ Base in percentage points; † marks differences whose 95% t-interval ($\Delta \pm 1.242$ SD) excludes zero. Bold marks the largest mean in each Base or Variant row; — marks a missing five-seed base result.

	MABC	**ITD3+BC**	**ICQL**	**MABCQ**	**IICQ**	**OMIGA**	**InSPO**
Lift dual *(Guided pool)*							
Base	**100.0** ±0.0	69.2 ±38.7	57.2 ±32.0	99.2 ±0.8	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
Variant	**100.0** ±0.0	0.0 ±0.0	88.4 ±9.2	73.0 ±3.7	99.8 ±0.4	96.0 ±3.4	98.8 ±1.1
Δ	0.0 ±0.0	$-69.2^{\dagger}$ ±38.7	$+31.2$ ±30.5	$-26.2^{\dagger}$ ±4.0	-0.2 ±0.4	-4.0 ±3.4	-1.2 ±1.1
Lift dual *(Mixed pool)*							
Base	43.0 ±8.9	79.2 ±8.9	10.2 ±20.6	44.4 ±5.1	41.0 ±6.1	**100.0** ±0.0	**100.0** ±0.0
Variant	12.6 ±8.1	24.0 ±9.2	30.6 ±6.2	0.8 ±1.3	12.6 ±4.3	**98.4** ±3.6	96.6 ±1.1
Δ	$-30.4^{\dagger}$ ±14.8	$-55.2^{\dagger}$ ±7.7	$+20.4$ ±21.0	$-43.6^{\dagger}$ ±5.9	$-28.4^{\dagger}$ ±8.6	-1.6 ±3.6	$-3.4^{\dagger}$ ±1.1
Push dual *(Guided pool)*							
Base	97.8 ±1.3	21.2 ±23.6	81.6 ±12.6	—	96.8 ±1.8	**98.0** ±1.4	93.2 ±11.4
Variant	96.2 ±1.5	38.4 ±24.4	82.2 ±6.2	—	96.0 ±2.9	97.0 ±2.0	**97.2** ±2.4
Δ	-1.6 ±1.3	$+17.2$ ±41.5	$+0.6$ ±17.0	—	-0.8 ±2.2	-1.0 ±2.2	$+4.0$ ±13.2

continued on next page

Table 13 continued

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Push dual *(Mixed pool)*							
Base	33.2 ±5.8	76.2 ±16.4	77.4 ±4.0	44.2 ±13.0	38.4 ±5.3	**96.4** ±2.3	76.6 ±29.8
Variant	36.8 ±3.6	77.2 ±11.5	79.4 ±8.0	35.8 ±3.0	38.8 ±6.1	94.0 ±4.9	**97.2** ±1.8
Δ	+3.6 ±7.3	+1.0 ±22.6	+2.0 ±8.1	−8.4 ±15.7	+0.4 ±3.8	−2.4 ±3.4	+20.6 ±29.8
Handover dual *(Guided pool)*							
Base	99.4 ±1.3	13.0 ±15.7	1.2 ±2.7	7.0 ±6.0	**99.8** ±0.4	99.0 ±1.0	65.6 ±30.4
Variant	99.6 ±0.5	2.4 ±3.4	0.0 ±0.0	35.6 ±7.7	99.6 ±0.9	**100.0** ±0.0	95.2 ±4.4
Δ	+0.2 ±1.6	−10.6 ±17.0	−1.2 ±2.7	+28.6† ±12.4	−0.2 ±1.1	+1.0 ±1.0	+29.6 ±31.3
Handover dual *(Mixed pool)*							
Base	68.2 ±8.4	72.0 ±6.7	0.0 ±0.0	10.0 ±13.2	71.6 ±9.1	**97.8** ±2.3	59.6 ±12.7
Variant	57.6 ±7.4	52.2 ±10.2	0.0 ±0.0	2.8 ±6.3	57.4 ±7.8	**93.8** ±7.7	22.8 ±20.1
Δ	−10.6 ±11.2	−19.8† ±8.0	0.0 ±0.0	−7.2 ±16.0	−14.2† ±8.9	−4.0 ±8.5	−36.8† ±21.2
Insert dual *(Guided pool)*							
Base	97.4 ±2.2	0.0 ±0.0	25.6 ±17.9	65.0 ±8.1	**97.8** ±1.8	93.6 ±4.7	90.0 ±8.2
Variant	61.2 ±5.6	0.0 ±0.0	0.8 ±1.8	58.4 ±12.6	65.8 ±6.6	**68.4** ±11.4	49.4 ±18.6
Δ	−36.2† ±4.1	0.0 ±0.0	−24.8† ±17.4	−6.6 ±6.5	−32.0† ±6.1	−25.2† ±9.8	−40.6† ±22.1
Insert dual *(Mixed pool)*							
Base	14.6 ±4.6	7.0 ±3.7	10.0 ±7.9	1.0 ±1.2	12.2 ±2.7	**93.2** ±3.8	82.2 ±4.3
Variant	3.0 ±0.7	0.8 ±1.3	0.2 ±0.4	6.8 ±2.2	2.2 ±1.3	**44.2** ±9.9	25.4 ±8.0
Δ	−11.6† ±4.9	−6.2† ±3.4	−9.8† ±7.6	+5.8† ±2.6	−10.0† ±1.6	−49.0† ±12.8	−56.8† ±9.0

Table 14: **ActiveSuction versus the base task.** Same conventions as Table 13.

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
Lift dual *(Guided pool)*							
Base	**100.0** ±0.0	69.2 ±38.7	57.2 ±32.0	99.2 ±0.8	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
Variant	98.8 ±2.2	0.0 ±0.0	76.2 ±24.4	84.6 ±6.8	99.2 ±1.1	**99.4** ±1.3	99.0 ±2.2
Δ	−1.2 ±2.2	−69.2† ±38.7	+19.0 ±49.2	−14.6† ±6.3	−0.8 ±1.1	−0.6 ±1.3	−1.0 ±2.2
Lift dual *(Mixed pool)*							
Base	43.0 ±8.9	79.2 ±8.9	10.2 ±20.6	44.4 ±5.1	41.0 ±6.1	**100.0** ±0.0	**100.0** ±0.0
Variant	27.0 ±8.6	2.0 ±1.2	23.2 ±19.7	5.2 ±4.0	16.4 ±3.6	**97.8** ±2.9	90.6 ±3.4
Δ	−16.0† ±8.8	−77.2† ±8.6	+13.0 ±25.3	−39.2† ±5.1	−24.6† ±3.4	−2.2 ±2.9	−9.4† ±3.4
Handover dual *(Guided pool)*							
Base	99.4 ±1.3	13.0 ±15.7	1.2 ±2.7	7.0 ±6.0	**99.8** ±0.4	99.0 ±1.0	65.6 ±30.4
Variant	81.2 ±11.0	2.0 ±3.4	0.4 ±0.9	33.0 ±18.6	92.8 ±8.3	**98.2** ±2.0	53.6 ±48.5
Δ	−18.2† ±10.0	−11.0 ±17.0	−0.8 ±3.0	+26.0† ±20.2	−7.0 ±7.9	−0.8 ±2.3	−12.0 ±66.3
Handover dual *(Mixed pool)*							
Base	68.2 ±8.4	72.0 ±6.7	0.0 ±0.0	10.0 ±13.2	71.6 ±9.1	**97.8** ±2.3	59.6 ±12.7
Variant	58.4 ±13.1	23.4 ±21.4	0.0 ±0.0	5.8 ±4.9	48.4 ±33.0	**76.4** ±25.1	32.6 ±27.4
Δ	−9.8 ±18.1	−48.6† ±21.4	0.0 ±0.0	−4.2 ±15.7	−23.2 ±28.3	−21.4 ±26.2	−27.0 ±25.1
Insert dual *(Guided pool)*							
Base	97.4 ±2.2	0.0 ±0.0	25.6 ±17.9	65.0 ±8.1	**97.8** ±1.8	93.6 ±4.7	90.0 ±8.2
Variant	**94.8** ±2.5	0.0 ±0.0	27.4 ±29.1	65.4 ±9.8	90.4 ±3.3	86.4 ±6.3	90.0 ±7.3
Δ	−2.6 ±3.6	0.0 ±0.0	+1.8 ±35.2	+0.4 ±16.2	−7.4† ±4.0	−7.2 ±6.5	0.0 ±7.3
Insert dual *(Mixed pool)*							
Base	14.6 ±4.6	7.0 ±3.7	10.0 ±7.9	1.0 ±1.2	12.2 ±2.7	**93.2** ±3.8	82.2 ±4.3
Variant	19.0 ±10.2	4.8 ±4.1	6.0 ±9.8	3.0 ±2.3	26.2 ±7.7	**75.6** ±9.7	67.8 ±8.8
Δ	+4.4 ±13.6	−2.2 ±7.4	−4.0 ±6.3	+2.0† ±1.6	+14.0† ±7.0	−17.6† ±9.7	−14.4† ±6.0

A.10 COMPETITIONBUTTON: TASK SUCCESS

CompetitionButton scores zero for every method. This value is not a missing result; it follows from the competitive success criterion. Figure 13 shows an MABC checkpoint trained on the Good pool: the right arm, controlled by Agent1, learns to complete the task, whereas Agent2 acquires no effective behavior. Against the fixed random opponent, the policy wins 67% of episodes in the Agent1 seat and none in the Agent2 seat, so $W_{\min} = 0\%$. A criterion that counts an episode as a success whenever either side wins would rate this checkpoint as competent. The zero therefore records that one seat dominates, which is the case $W_{\min}$ is designed to separate from a contested outcome.

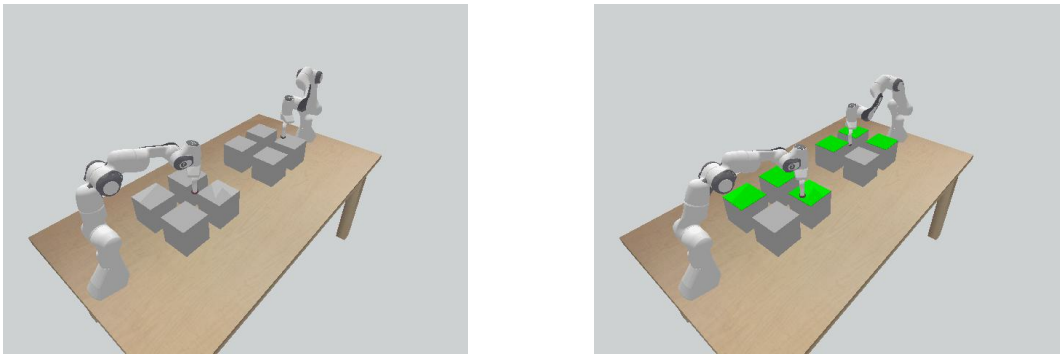


Figure 13: **CompetitionButton Good MABC rendering.** Initial (left) and terminal (right) frames from the first unfiltered episode. Agent1 (right arm) wins at step 8.

A.11 SENSITIVITY TO A FIXED LEARNED OPPONENT

All competitive results in the main text are measured against the same random opponent. Combined with $W_{\min}$, which requires a policy to win from both seats, this evaluation reflects competitive learning to some extent and exposes cases in which one agent has learned while the other has not. It also places all methods on a common reference. It does not show, however, how the results change when the opponent is itself trained. We therefore run a check on three MuJoCo Point races. For each task, the Final checkpoints of all seven methods (four training seeds each, trained on the 50k Good pool) play against the random opponent and against a fixed MABC policy trained on the same pool with a different seed. Each policy plays 100 episodes in each seat against each opponent, and we take the smaller of its two seat win rates as $W_{\min}$ before averaging over the four seeds. On the two reach races, $W_{\min}$ drops substantially against the learned opponent, and the best method changes from OMIGA to ICQL. On PointPushRace, $W_{\min}$ changes little. Overall, the method ranking is largely preserved (Kendall $\tau_b = 0.78$, 0.59, and 0.98; Table 15). The random-opponent results in the main text should therefore be read as a common reference for comparing methods, not as a measure of performance against competent opponents.

A.12 STABILITY OF THE FAMILY-LEVEL COMPARISON

Figure 4 already reports method performance at the family level. Here we examine how consistent the resulting rankings are across families and how much they depend on the data budget. We consider only Panels A and B: for each family, we average the five-seed Final means of its tasks with equal weight and rank the seven methods. The mean Kendall τ_b (Kendall et al., 1938) over the 21 pairs of families is 0.33 under Good data and 0.39 under Medium data (95% intervals $[0.29, 0.43]$ and $[0.33, 0.41]$, obtained by resampling the five training seeds). Rankings therefore differ considerably between families, which is why Figure 4A–B reports them per family rather than as a single aggregate score. The rankings also share common elements: under Medium data, OMIGA leads in five of the seven families.

To reduce the influence of the data budget on the family comparison, we restrict the main results to tasks with 50k pools. This removes the 22 MuJoCo Panda and Point tasks with 500k pools and leaves 34 of the 56 tasks (Table 16). Under **Medium** data, the best method is unchanged in every

Table 15: $W_{\min}$ **against the random and a fixed learned opponent.** Final checkpoints trained on the 50k Good pool of each task; mean with sample standard deviation over four training seeds, all facing the same learned MABC opponent. Bold marks the largest mean in a row; τ_b compares the method rankings under the two opponents.

	MABC	ITD3+BC	ICQL	MABCQ	IICQ	OMIGA	InSPO
PointReachRace *($\tau_b = 0.78$)*							
Random	97.3 ±0.5	52.5 ±3.0	97.5 ±0.6	76.3 ±7.1	**98.0** ±0.8	**98.0** ±0.0	93.3 ±3.4
Learned	33.8 ±1.3	13.0 ±2.4	**43.8** ±1.3	17.5 ±3.1	35.3 ±0.5	41.0 ±0.0	32.3 ±2.6
PointReachAvoidRace *($\tau_b = 0.59$)*							
Random	93.8 ±0.5	39.0 ±4.7	84.5 ±5.1	62.5 ±2.5	93.8 ±0.5	**94.0** ±0.0	78.3 ±2.9
Learned	37.8 ±0.5	11.5 ±3.7	**46.3** ±2.8	11.3 ±3.9	38.0 ±2.4	44.3 ±0.5	28.8 ±1.7
PointPushRace *($\tau_b = 0.98$)*							
Random	**31.8** ±3.5	1.5 ±1.3	0.8 ±1.5	0.5 ±0.6	31.3 ±3.0	10.5 ±2.1	22.8 ±4.6
Learned	**37.0** ±1.6	1.5 ±1.3	1.0 ±0.8	1.0 ±1.4	35.8 ±2.8	10.3 ±2.6	28.0 ±4.7

family except *Competition*, although the full ranking changes more in *Multi-Goal* and *Competition* ($\tau_b = 0.24$ and 0.14). Under **Good** data, where most methods perform well, the best method changes in four of the seven families.

Table 16: **Family winners without the 500k tasks.** Best method by task-equal family mean over all 56 tasks and over the 34 tasks with 50k pools (an arrow marks a change); τ_b compares the two rankings of the seven methods. *MABC, MABCQ, IICQ, and OMIGA tie.

		Good		Medium	
Family	**Tasks (all / 50k)**	Best	τ_b	Best	τ_b
Cooperation	19 / 12	IICQ → OMIGA	0.78	OMIGA	0.81
Transition	10 / 8	OMIGA	1.00	OMIGA	0.88
Multi-Goal	9 / 4	OMIGA → ICQL	0.05	OMIGA	0.24
Hybrid	2 / 1	OMIGA	0.90	ITD3+BC	0.90
Intervention	5 / 4	InSPO	1.00	OMIGA	0.90
Competition	9 / 4	ITD3+BC → InSPO	0.52	ITD3+BC → InSPO	0.14
Trajectory-Tracking, one-shot	2 / 1	OMIGA → tie*	0.70	OMIGA	0.90

A.13 Single-Agent Online and Offline Extensions

Here, we present part of the training results using single-agent offline data (Table 17), and also single-agent online results (Figure 14). SAC+HER (Haarnoja et al., 2018; Andrychowicz et al., 2018) is implemented for online results. The single-policy experiments assess whether the platform also supports conventional centralized-control studies alongside decentralized multi-agent evaluation.

Table 17: **Single-agent offline learning.** Success rate (%) is reported as mean ± sample standard deviation over five seeds; each pool contains 50,000 transitions. Bold marks the best method per task. Final is the fixed-endpoint checkpoint; Best is selected on clean validation.

	Stack (dual)		Lift (dual)		Push (dual)	
Method	Final	Best	Final	Best	Final	Best
BCQ	**100.0** ±0.0	**99.4** ±0.5	**100.0** ±0.0	**100.0** ±0.0	**98.2** ±1.3	**98.8** ±0.8
TD3+BC	1.4 ±1.9	93.6 ±2.7	69.4 ±23.1	98.4 ±0.9	58.8 ±19.3	62.2 ±7.7
CQL	0.0 ±0.0	57.4 ±9.7	33.2 ±39.6	97.4 ±1.8	1.2 ±2.7	85.6 ±6.9

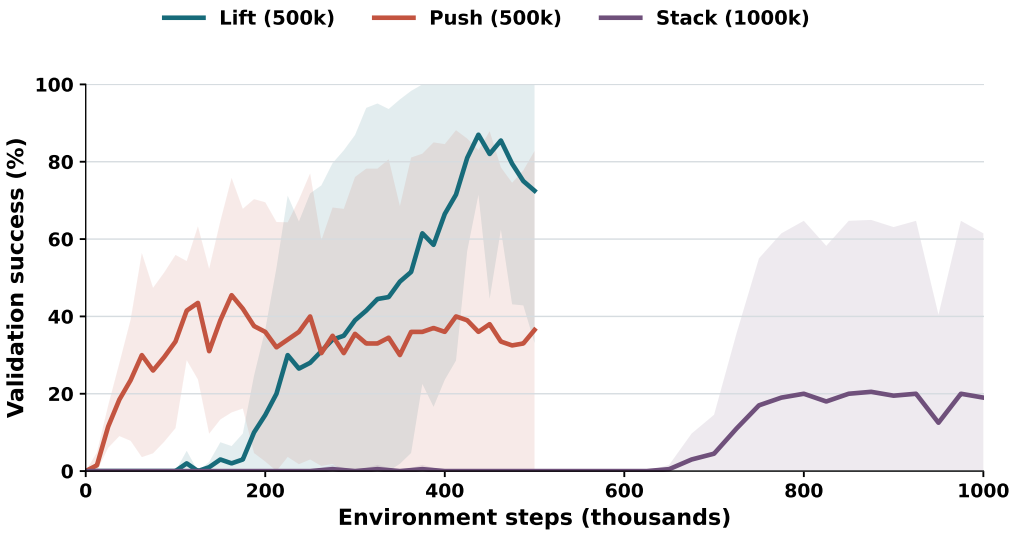


Figure 14: **PyBullet single-agent online learning**. Online learning curves are averaged over five seeds, with shaded regions indicating the sample standard deviation across seeds rather than the uncertainty of the mean.

A.14 Pose-Error Robustness

The pose-perturbation study uses the RELATIVE configuration on tasks where an arm interacts with an object, excluding discrete tasks, MuJoCo Point tasks, all *Trajectory-Tracking* tasks, and tasks without object interaction. Besides clean evaluation, we use three perturbation levels with standard deviations of 5 mm/2°, 10 mm/5°, and 20 mm/10° in position and orientation. Figure 15 reports Best-checkpoint results. Across the 34 eligible success-rate tasks under **Good**, success decreases monotonically with the perturbation level for every method. The loss is largest for the strongest methods: MABC falls from 88.5% to 49.4% and OMIGA from 88.0% to 51.3%, and IICQ follows a similar curve. InSPO, ITD3+BC, ICQL, and MABCQ start from lower clean success and lose fewer points. Under **Medium**, OMIGA and InSPO remain the strongest methods at every perturbation level. On the two **Medium** *Competition* tasks, $W_{\min}$ changes little, and OMIGA, IICQ, and MABC even rise slightly.

A.15 Collision

The safety analysis considers eight close-proximity dual-arm tasks: Lift dual, Insert dual, Connect dual, and Handover dual in PyBullet, and DualLift, Handover, SlideHandover, and Stack in MuJoCo Panda. In these tasks, the two arms operate in proximity, which makes inter-arm collisions more likely and can terminate an episode. A clearance violation is recorded when the minimum surface distance between the two arms falls below 1 cm; normal arm–object contact is excluded. We report both episode-level and step-level violation rates, together with task success without any violation (Figure 16). Because a low violation rate may also result from limited effective motion, it should be interpreted jointly with task completion. Clearance is checked at environment-step boundaries in PyBullet and at physics substeps in MuJoCo.

A.16 Final and Best Checkpoints

Training exposes a late-stage degradation in some methods, where the Final weights fall well short of the best validated checkpoint (Figure 17); InSPO and ITD3+BC show it most clearly. We quantify it by comparing **Final** against **Best** on the 45 tasks whose native metric is success rate, leaving aside the discrete and competitive tasks, which are scored on other quantities. Across those tasks OMIGA

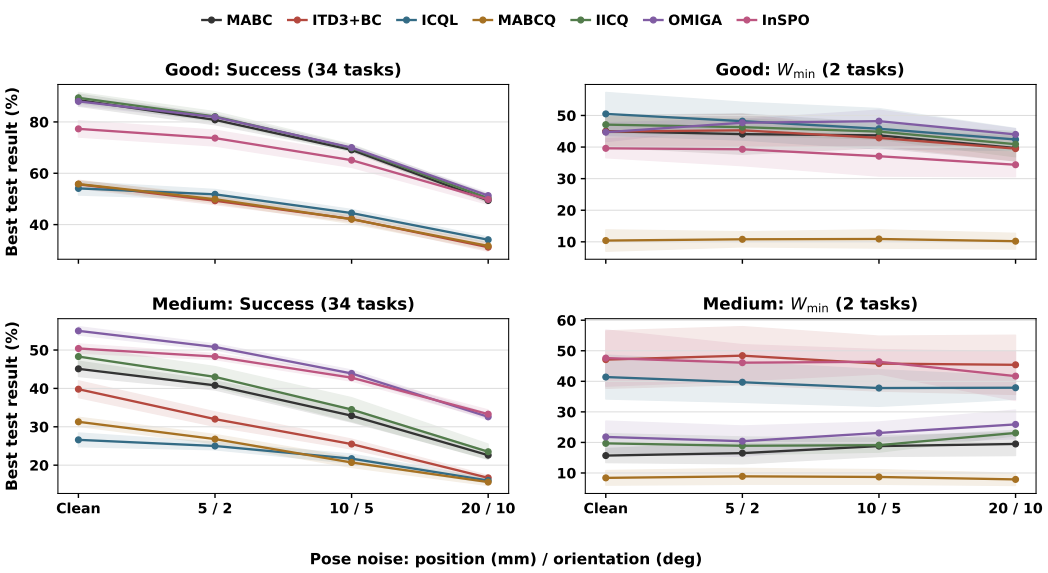


Figure 15: **Best-checkpoint pose-error evaluation.** Rows show Good and Medium data; columns show success on the 34 eligible tasks and $W_{\min}$ on the two *Competition* tasks. The horizontal axis gives the position (mm) and orientation (deg) noise level. Curves average the common eligible tasks; bands show sample standard deviations across five training seeds.

carries the smallest gap. Unless stated otherwise, results throughout the paper are reported at **Final**.

A.17 Data-Scale Comparison

The main results use 500k-transition pools for the MuJoCo Panda and Point tasks, whose horizons are generally longer, and 50k pools elsewhere, whereas the eight-construction comparison in the main text uses 50k transitions throughout to keep the data budget matched across constructions. To examine the effect of this choice, we compare the MuJoCo Panda and Point tasks under 50k and 500k budgets (Figure 18).

Increasing the dataset size generally improves performance, most markedly on the Point tasks, while for some tasks, such as the Soccer tasks, the difference between the two scales is small. Because the data-composition study compares constructions at a common 50k budget, this scale effect shifts absolute performance on MuJoCo tasks rather than explaining the differences between constructions; the absolute MuJoCo numbers in that study should nonetheless be read as lower than at the main-result scale.

A.18 Independent Adaptations of OMIGA and InSPO

OMIGA and InSPO were originally proposed for cooperative settings with a shared team reward. Neither objective is defined when the agents' rewards are opposed, as in the *Competition* family. We therefore train all methods under the same Independent configuration, so that every method receives the same actor inputs and learning targets on all seven families. On our tasks, centralized training does not give a consistent advantage either; in Appendix A.19 it improves MATD3+BC but not MACQL.

In the Independent configuration, each agent is trained on its own inputs and targets. For OMIGA, we instantiate one single-agent learner per role. An adapter maps the local observation and goal of that role to the state fields of the learner and passes only the role's own action, reward, next local fields, and terminal flag. The value module thus has no access to peer actions or the global state and cannot mix values across agents. For InSPO, each role has its own actor, behavior model, and critic. The critic is conditioned on the observation and goal of the role, and its target is computed

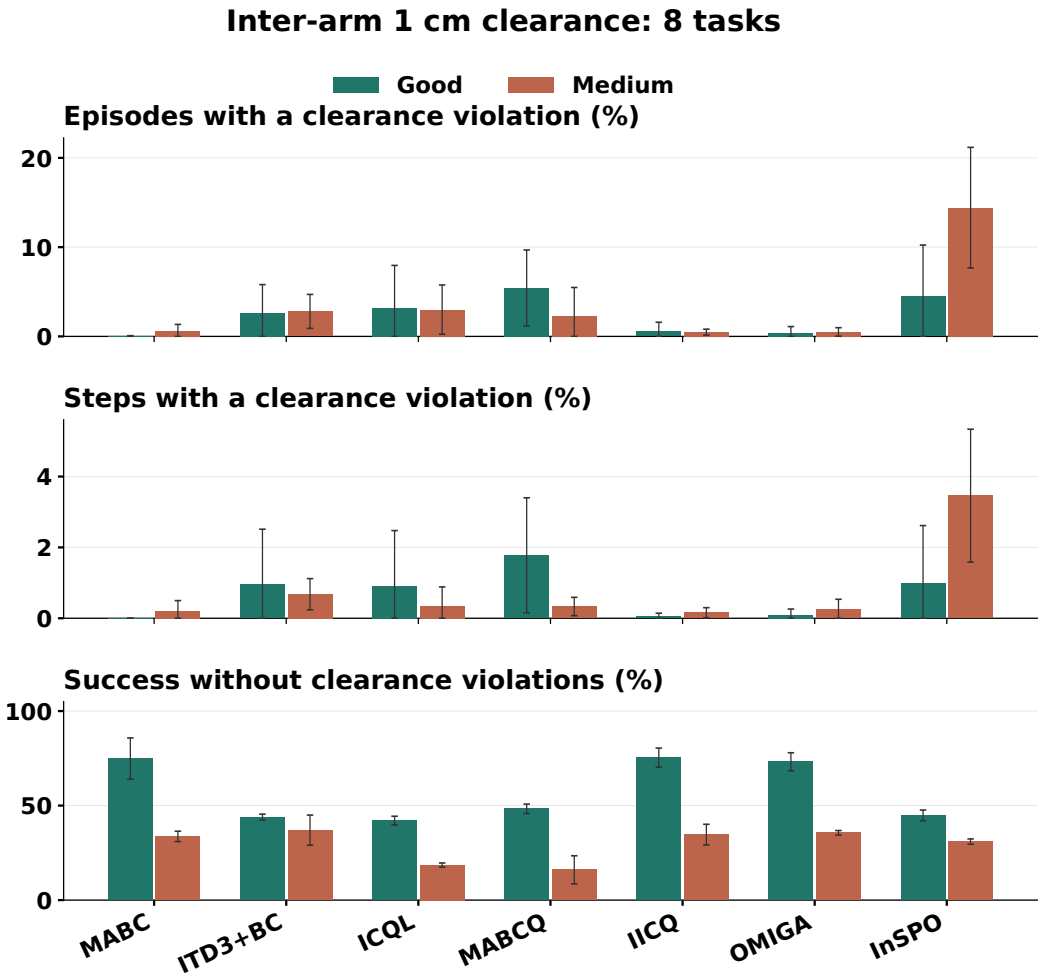


Figure 16: **Violation evaluation on the eight prespecified close-proximity tasks.**

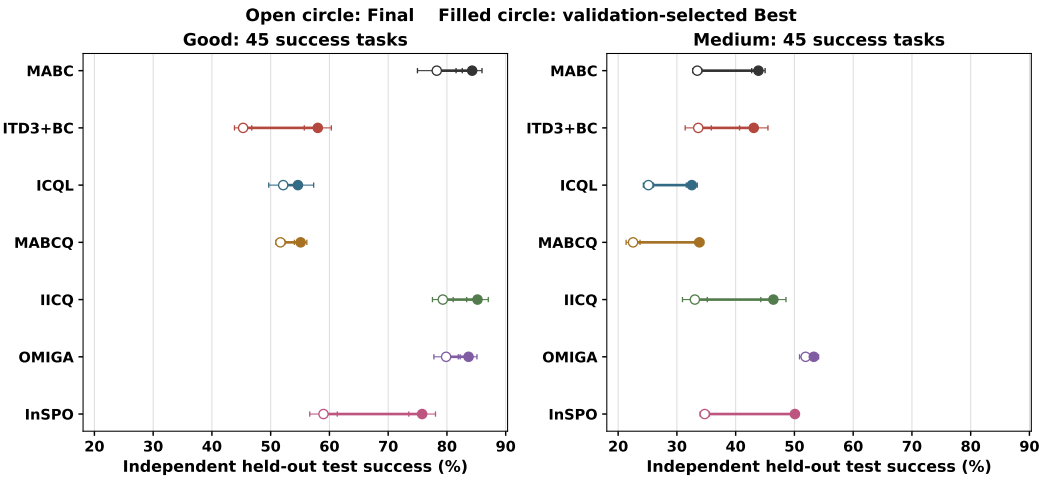


Figure 17: **Final versus validation-selected Best results on matched conditions.**

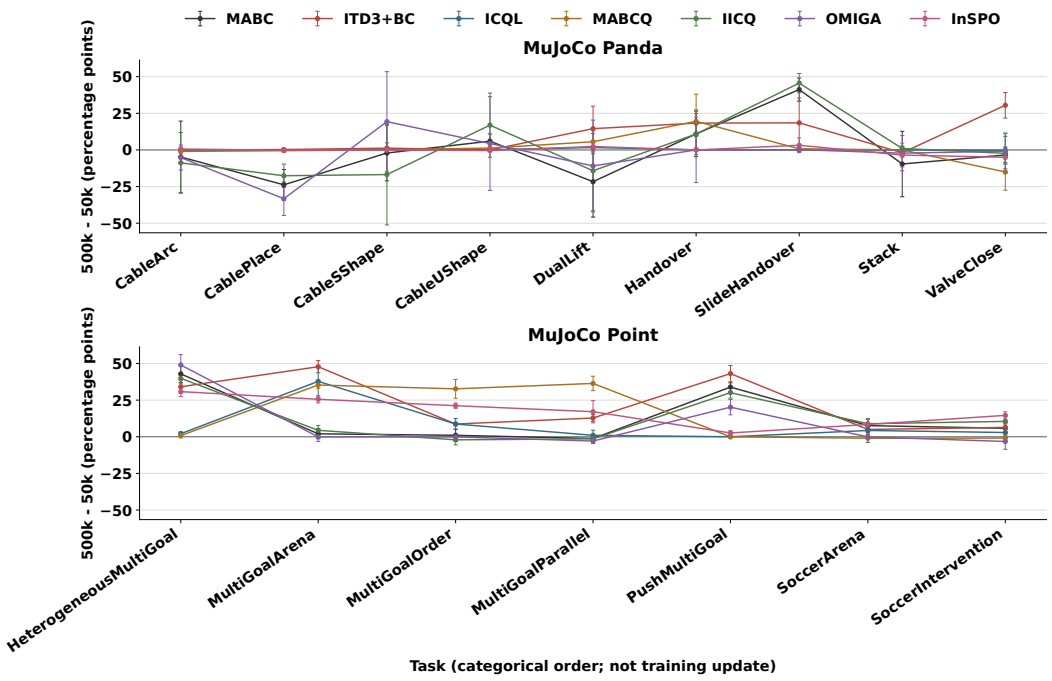


Figure 18: **Task- and source-paired Final differences between 500k and 50k datasets.** Differences are 500k minus 50k, in percentage points, for the MuJoCo Panda (top) and Point (bottom) tasks; markers are methods.

from the role's reward and terminal flag. In contrast to the CTDE version, this variant does not fit a conditional density over teammate actions and does not resample with teammate importance ratios. We keep the per-agent behavior regularization and the sequential update loop, although sequential updates alone do not make the objective depend on teammates. At test time, the independently trained actors act together in the shared environment.

In the rest of the paper, OMIGA and InSPO refer to these local-objective adaptations, and the reported results should be read as results for the adaptations, not for the original coupled objectives.

A.19 Independent versus Centralized-Training Configurations

The main results use the Independent configuration, following the recommendations of Pan et al. (2022) and Barde et al. (2024). We also compare CTDE against Independent training for MACQL and MATD3+BC, whose Independent forms appear in the main text as ICQL and ITD3+BC, on the 50k and 500k collections (Table 18). Centralized training brings no consistent benefit (Figure 19): across the 50k matched conditions, it improves MATD3+BC but not MACQL, and the smaller 500k cohort shows no stable gain.

A.20 Reference Action and State–Action Coverage

Coverage is measured against a reference behavior rather than against the reachable space. Candidates are the Good, Medium, and Poor pools of the main results, constructed as in Section 4.1; within a task, the Good source supplies episode-disjoint calibration and query sets, and copied reference episodes are dropped from the candidates. For a representation ϕ, standardization and a 95%-variance PCA (Jolliffe & Cadima, 2016) are fitted on the calibration data, and the query-specific fifth-neighbor radius $r_5(q)$ gives

$$C(D;Q) = \frac{100}{|Q|} \sum_{q \in Q} \mathbf{1}\left[\min_{x \in D} \|\phi(q) - \phi(x)\|_2 \le r_5(q)\right]. \tag{8}$$

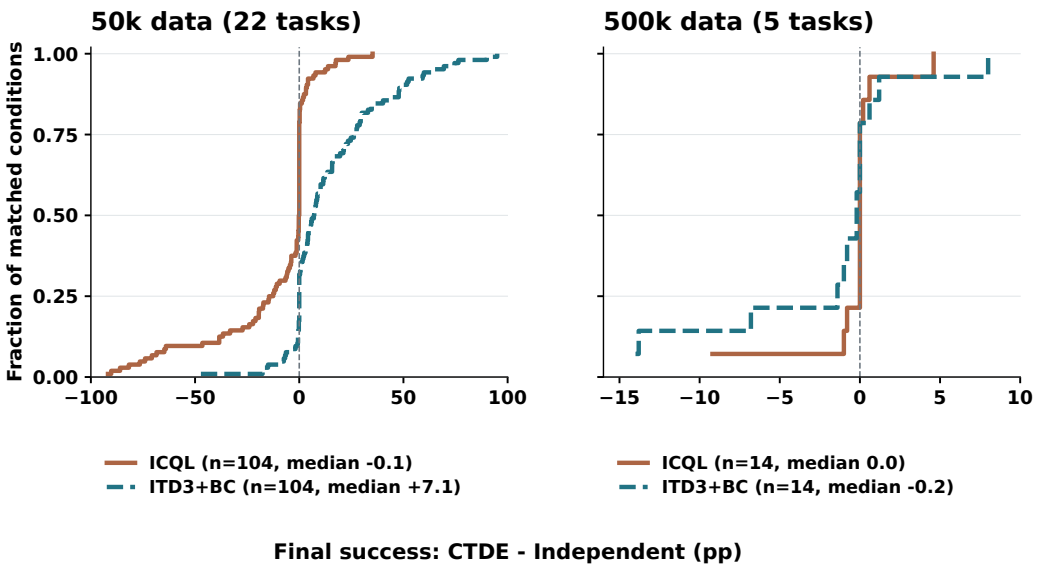


Figure 19: **Paired CTDE effects under matched data conditions.** Curves are empirical cumulative distributions of $\Delta_c = \bar{s}_c^{\mathrm{CTDE}} - \bar{s}_c^{\mathrm{Independent}}$, with $\bar{s}$ the Final success rate averaged over matched training seeds. A condition c holds the task, collection source, actor input, physical dataset, and transition budget fixed, so Δ_c varies the training configuration and nothing else. A positive Δ_c thus favors CTDE as a whole configuration: for MACQL, the CTDE setting also rescales own-action regularization, which Δ_c cannot separate from the change in critic input. Legends report n, the number of paired conditions per method, and the median Δ_c over those conditions in percentage points; neither is a task-equal or seed-level statistic. Each method contributes $n = 104$ conditions across 22 tasks at 50k and $n = 14$ across five tasks at 500k.

Table 18: **Task support for the aligned CTDE comparison.** Registry prefixes are omitted. Bold tasks also enter the 500k panel.

Platform	Tasks	50k	500k
PyBullet	Asyn dual; Connect dual; Handover dual and triple; HandoverActiveSuction dual; HandoverIntervention dual; HandoverPush dual, triple, and quad; Insert dual; InsertActiveSuction dual; InsertIntervention dual; Lift dual; PushButton dual; ReachSequential dual	15	0
MuJoCo	DualGripperHandoverProxy; DualGripperValveTurnCloseProxy; **PandaCablePlace**; **PandaHandover**; **PandaValveClose**; **PointMultiGoalArena**; **PointSoccerIntervention**	7	5
Total		**22**	**5**

Four representations are evaluated, each standardized and projected independently:

$$\begin{aligned} \phi_J^A(t) &= (a_t^1, \ldots, a_t^N), & \phi_i^A(t) &= a_t^i, \\ \phi_J^{SA}(t) &= (s_t, a_t^1, \ldots, a_t^N), & \phi_i^{SA}(t) &= (o_t^i, g_t^i, a_t^i), \end{aligned} \tag{9}$$

with s_t the central state, o_t^i the local observation, and g_t^i the local goal. Candidates are subsampled to at most $2{,}048$ transitions, and the calibration and query sets to $1{,}024$ and 512; each entry averages five such resamples, which measure sampling variability rather than training-seed variability.

What the two tables show. Tables 19 and 20 score every entry relative to the Good data of the same task. The two tables ask whether the candidate data contain similar actions, and whether those actions occur near similar states; within each table, the per-agent and joint columns ask whether each agent's actions are supported individually, and whether their synchronous combination is supported. The Medium and Poor pools expose the limits of the quality label. On the discrete tasks, for example, Poor attains full or near-full action coverage—the legal action set is small enough for random behavior to exhaust it—yet its joint state–action coverage falls to between 14.3 and 49.5 (Table 20): these actions are mostly not taken in the states where they matter. The support of offline multi-agent data therefore cannot be summarized by action diversity or by the Good/Medium/Poor label alone; it must also distinguish the state context in which actions occur, the local support of each agent, and the support of the joint action.

A.20.1 Action-Only Coverage

Table 19: **Joint and per-agent action coverage (%).** Each cell gives the Good / Medium / Poor means over five resamples for the pools used in the main results. C_J scores the joint action and C_i agent i's action, both against an episode-disjoint Good reference.

Task	C_J	C_1	C_2	C_3	C_4
PyBullet					
Asyn dual	98.4/30.7/30.5	98.3/43.2/35.7	97.9/58.6/46.9	—	—
CompetitionButton dual	95.4/52.1/58.9	97.5/68.9/73.1	97.0/65.7/62.5	—	—
CompetitionPush dual	98.8/67.1/67.3	99.3/84.8/81.1	98.8/88.3/86.2	—	—
Connect dual	98.3/75.3/49.0	98.4/85.2/65.0	98.2/83.1/55.4	—	—
Curling dual	97.6/67.5/63.2	98.8/79.3/74.1	98.8/87.6/77.8	—	—
GOODLettersRepeat quad	98.2/45.9/0.7	97.3/54.5/24.5	97.5/52.7/21.1	97.4/54.7/18.9	98.9/54.5/20.2
HILetters dual hiMulti	99.0/85.5/1.7	99.5/87.9/5.5	99.3/84.8/4.4	—	—
HILettersRepeat dual hiMulti	98.6/30.7/30.9	98.7/45.2/44.3	98.1/48.9/53.4	—	—
Handover dual	99.6/51.8/47.5	99.6/71.9/61.8	99.5/76.6/59.5	—	—
Handover triple	99.8/98.8/0.8	99.8/98.4/37.0	99.8/98.9/11.6	99.6/98.9/24.8	—
HandoverActiveSuction dual	99.7/98.6/0.0	99.4/98.1/3.9	99.8/98.9/3.8	—	—
HandoverIntervention dual	99.0/98.8/13.1	99.5/98.4/27.1	99.4/98.3/44.0	—	—

continued on next page

Table 19 continued

Task	C_J	C_1	C_2	C_3	C_4
HandoverPush dual	99.1/46.8/51.3	99.2/65.6/63.4	99.5/57.7/63.6	—	—
HandoverPush quad	99.6/96.6/0.7	99.5/97.6/24.1	99.4/97.6/21.6	99.3/97.6/25.2	99.6/97.6/22.2
HandoverPush triple	99.4/52.0/51.7	99.5/88.8/76.6	99.3/82.5/74.6	99.6/92.0/75.3	—
Insert dual	97.8/84.6/56.8	98.7/92.0/72.0	98.9/93.8/75.6	—	—
InsertActiveSuction dual	98.9/79.1/51.6	99.2/90.1/61.9	98.4/89.3/72.1	—	—
InsertIntervention dual	99.0/87.6/61.3	98.4/94.4/73.3	99.5/92.9/78.0	—	—
Lift dual	98.3/56.2/58.0	99.4/68.1/69.2	99.0/78.6/62.9	—	—
Lift quad	97.7/21.5/25.1	97.5/39.9/33.0	99.2/29.0/34.2	98.0/24.0/31.5	98.0/48.8/34.5
Lift triple	96.8/14.6/15.8	97.2/27.8/25.8	95.9/22.1/24.3	96.8/25.7/38.9	—
LiftActiveSuction dual	99.5/96.6/2.2	99.1/96.6/6.8	99.3/96.8/3.5	—	—
LiftIntervention dual	98.3/93.9/9.7	98.3/94.2/19.8	98.8/94.3/22.6	—	—
Push dual	98.1/29.1/33.0	97.9/44.5/45.6	97.6/42.4/40.9	—	—
PushButton dual	99.1/14.5/15.1	99.5/23.6/32.3	99.2/33.5/34.6	—	—
PushIntervention dual	97.0/27.6/18.5	97.4/46.1/26.8	97.4/42.9/29.8	—	—
PushSequential dual	99.5/82.2/38.7	99.6/92.6/67.2	99.7/94.7/67.3	—	—
Reach dual	92.9/77.0/35.8	95.7/75.3/53.5	93.4/75.8/54.7	—	—
ReachAvoid dual	96.3/24.9/14.7	97.3/44.8/26.7	98.0/42.4/39.6	—	—
ReachSequential dual	97.4/94.2/73.8	98.8/95.5/81.4	99.2/92.5/82.0	—	—
Soccer dual	99.4/23.5/23.3	99.6/56.1/51.2	99.5/54.8/48.2	—	—
Stack dual	97.7/52.4/31.7	97.7/64.0/43.9	97.7/76.2/58.4	—	—
TicTacToe dual	99.8/98.0/8.1	99.6/98.7/25.7	99.3/98.7/56.5	—	—
TrajectoryMPG triple	99.3/78.4/1.8	99.6/80.3/14.2	99.1/79.6/14.5	98.9/80.6/15.9	—
WindowsMultiGoal dual	99.2/65.3/40.0	99.3/83.9/67.7	99.2/81.6/62.0	—	—
MuJoCo					
DualGripperHandoverProxy	99.7/10.8/10.8	99.6/11.1/12.7	99.7/11.3/12.0	—	—
DualGripper ValveTurnCloseProxy	99.6/20.5/8.9	99.8/20.0/8.9	99.5/19.8/8.9	—	—
PandaCableArc	99.5/75.8/1.0	99.5/87.9/7.5	99.8/88.0/8.8	—	—
PandaCablePlace	99.4/63.2/1.2	99.8/77.1/5.2	99.7/75.1/4.8	—	—
PandaCableSShape	99.7/70.0/0.1	99.5/82.9/6.2	99.7/87.9/5.9	—	—
PandaCableUShape	99.6/98.6/1.3	99.6/98.8/9.0	99.7/99.0/5.1	—	—
PandaDualLift	99.7/73.4/1.1	99.7/80.4/11.2	99.6/91.6/12.3	—	—
PandaHandover	99.1/89.7/10.4	99.6/96.1/19.1	99.5/94.6/16.6	—	—
PandaSlideHandover	99.4/99.1/0.1	99.8/99.1/2.4	99.8/99.2/2.9	—	—
PandaStack	99.5/62.8/0.0	99.6/78.0/5.8	99.6/78.0/6.1	—	—
PandaValveClose	99.5/91.2/21.7	99.2/95.0/24.6	99.3/94.4/27.9	—	—
PointHeterogeneous MultiGoal	97.0/85.1/40.6	99.4/92.6/47.7	99.5/92.8/68.8	—	—
PointMaze3v1Arena	84.7/81.2/81.1	93.8/79.6/72.7	91.2/80.5/76.7	90.4/82.1/77.4	95.3/88.0/82.9
PointMaze3v1Medium	84.4/73.9/70.0	98.2/46.0/27.4	98.3/40.4/22.0	98.2/42.5/28.0	98.3/47.8/26.9
PointMultiGoalArena	97.8/86.2/58.6	99.4/91.8/59.4	99.0/94.2/63.9	—	—
PointMultiGoalOrder	97.1/96.6/78.9	99.6/97.9/77.1	99.0/98.8/74.5	—	—
PointMultiGoalParallel	97.1/92.0/62.6	99.2/88.1/49.7	99.4/87.8/49.3	—	—
PointPushMultiGoal	98.1/97.3/88.5	98.4/98.5/80.0	99.1/96.6/82.5	—	—
PointPushRace	91.4/89.6/62.3	98.8/96.2/65.1	98.8/93.5/60.0	—	—
PointReachAvoidRace	88.4/93.2/58.8	93.6/84.0/47.8	93.4/84.8/45.3	—	—
PointReachRace	80.9/66.6/35.3	96.4/75.1/32.5	97.3/80.4/31.8	—	—
PointSoccerArena	97.9/74.5/78.4	99.7/90.7/88.1	99.5/90.0/86.1	99.4/89.6/88.2	—
PointSoccerIntervention	98.9/76.6/79.8	99.8/87.9/83.2	99.5/87.4/85.9	99.6/87.5/83.4	—
PointTrajectoryHI	99.6/76.2/52.5	99.6/96.7/68.9	99.6/96.2/86.9	—	—

continued on next page

Table 19 continued

Task	C_J	C_1	C_2	C_3	C_4
Discrete					
CompetitiveButtonRace	100.0/100.0/ 100.0	100.0/100.0/ 100.0	100.0/100.0/ 100.0	—	—
KeyedHandoverProtocol	98.3/93.5/ 100.0	100.0/100.0/ 100.0	100.0/100.0/ 100.0	—	—
KeyedHandoverProtocol Competitive	100.0/77.7/ 97.4	100.0/97.7/ 100.0	100.0/100.0/ 100.0	—	—
KeyedHandoverProtocol MultiGoal	100.0/97.9/ 100.0	100.0/100.0/ 100.0	100.0/100.0/ 100.0	—	—
MixedCurling	100.0/99.3/ 100.0	100.0/100.0/ 100.0	100.0/100.0/ 100.0	—	—
Stack	100.0/94.3/ 100.0	100.0/100.0/ 100.0	100.0/100.0/ 100.0	—	—

A.20.2 State–Action Coverage

Table 20: **Joint and per-agent state–action coverage (%).** Conventions follow Table 19. C_J scores the central state with the joint action; C_i scores agent i's local observation, goal, and action. Each representation is calibrated against its own episode-disjoint Good reference.

Task	C_J	C_1	C_2	C_3	C_4
PyBullet					
Asyn dual	98.7/13.0/9.9	99.1/9.5/12.8	98.8/15.2/10.9	—	—
CompetitionButton dual	96.1/21.1/22.3	98.3/32.1/19.4	94.3/20.3/30.7	—	—
CompetitionPush dual	87.9/18.9/18.9	93.9/17.6/17.8	88.0/13.4/22.4	—	—
Connect dual	94.7/66.8/8.9	96.9/59.3/12.1	93.7/77.5/13.9	—	—
Curling dual	92.4/23.9/11.9	88.4/27.9/13.2	93.0/37.3/24.7	—	—
GOODLettersRepeat quad	98.7/46.2/0.6	98.6/41.8/0.7	97.9/43.2/0.7	98.4/43.0/0.6	98.9/44.2/0.7
HILetters dual hiMulti	99.2/86.1/0.4	91.8/58.6/0.7	94.4/76.8/0.6	—	—
HILettersRepeat dual hiMulti	98.9/7.9/0.3	98.7/5.5/0.4	97.3/3.8/0.3	—	—
Handover dual	99.3/14.3/14.9	97.9/20.8/19.9	97.8/25.9/15.5	—	—
Handover triple	99.4/97.6/1.4	99.2/97.6/3.8	98.8/96.4/3.6	98.5/96.4/2.7	—
HandoverActiveSuction dual	98.6/97.7/0.1	96.6/95.2/2.5	98.7/97.1/1.2	—	—
HandoverIntervention dual	98.3/96.7/3.4	95.2/94.8/8.0	97.2/97.6/5.2	—	—
HandoverPush dual	98.7/21.8/19.0	98.4/23.5/23.0	98.3/29.3/35.9	—	—
HandoverPush quad	99.6/96.9/0.1	99.5/95.9/3.8	99.3/97.0/4.0	99.8/97.0/3.6	99.2/97.3/3.9
HandoverPush triple	99.6/18.0/13.2	98.3/32.4/16.2	97.6/22.4/15.4	99.6/28.8/19.7	—
Insert dual	94.1/48.8/12.2	95.2/31.0/11.6	93.5/74.3/17.7	—	—
InsertActiveSuction dual	95.2/63.4/26.5	97.0/45.0/39.4	97.1/77.2/36.1	—	—
InsertIntervention dual	97.8/78.8/17.7	96.4/76.9/14.9	97.0/62.0/20.9	—	—
Lift dual	96.6/18.6/18.2	94.8/30.5/22.5	95.8/26.0/18.4	—	—
Lift quad	97.2/7.7/12.0	98.0/16.6/16.0	97.5/9.4/13.9	94.7/9.5/14.2	96.0/14.2/16.5
Lift triple	97.2/15.2/7.5	97.8/31.4/10.9	98.7/19.1/8.2	95.9/9.4/8.4	—
LiftActiveSuction dual	99.5/96.8/1.5	98.0/96.8/5.2	99.2/95.5/0.6	—	—
LiftIntervention dual	98.7/95.2/3.8	98.6/95.7/6.4	98.5/94.9/6.7	—	—
Push dual	98.0/12.7/12.2	96.8/20.2/17.3	97.0/18.9/16.5	—	—
PushButton dual	97.8/17.0/10.2	99.4/12.8/9.1	99.6/12.6/8.7	—	—
PushIntervention dual	96.6/11.1/11.3	95.8/14.4/18.5	95.0/15.6/20.3	—	—
PushSequential dual	98.9/76.1/9.8	98.7/75.1/11.1	98.9/72.7/11.2	—	—
Reach dual	90.3/88.7/15.4	93.7/88.3/21.3	94.3/88.2/23.6	—	—
ReachAvoid dual	97.3/27.4/6.2	97.3/21.7/8.0	98.3/14.2/9.9	—	—

continued on next page

Table 20 continued

Task	C_J	C_1	C_2	C_3	C_4
ReachSequential dual	98.1/88.1/26.9	93.7/82.3/22.0	96.8/74.5/17.1	—	—
Soccer dual	97.4/6.6/4.0	98.1/3.7/2.8	98.0/5.0/3.2	—	—
Stack dual	96.6/48.8/18.1	98.1/37.3/17.0	97.9/34.0/15.9	—	—
TicTacToe dual	99.6/98.1/2.1	99.7/98.1/6.4	99.5/98.3/5.3	—	—
TrajectoryMPG triple	99.5/77.5/0.6	98.6/77.7/0.6	99.1/78.5/0.5	99.2/76.4/0.5	—
WindowsMultiGoal dual	97.7/69.0/4.3	99.3/42.1/4.3	98.8/36.5/5.8	—	—
MuJoCo					
DualGripperHandoverProxy	99.8/1.6/1.8	99.7/2.4/2.2	99.7/1.4/2.0	—	—
DualGripper ValveTurnCloseProxy	99.6/28.6/2.1	99.6/25.9/4.4	99.5/27.0/4.8	—	—
PandaCableArc	98.8/9.8/0.5	98.4/11.1/0.7	98.9/10.0/0.6	—	—
PandaCablePlace	99.0/6.0/0.4	98.9/9.2/0.4	98.6/11.8/0.5	—	—
PandaCableSShape	99.3/6.7/1.0	99.7/9.9/0.7	99.6/8.9/1.0	—	—
PandaCableUShape	99.0/98.1/3.6	98.9/98.4/4.1	98.0/98.4/2.0	—	—
PandaDualLift	99.3/19.8/0.7	98.9/28.1/0.7	99.0/30.2/1.0	—	—
PandaHandover	94.6/65.0/1.8	95.9/75.0/5.1	95.5/65.3/2.3	—	—
PandaSlideHandover	98.9/99.0/0.2	99.7/99.4/0.4	99.4/99.4/0.2	—	—
PandaStack	98.6/14.2/0.1	99.6/18.3/0.3	99.6/9.1/0.1	—	—
PandaValveClose	98.3/84.1/0.9	99.1/86.9/1.1	98.9/87.1/2.0	—	—
PointHeterogeneous MultiGoal	87.1/90.9/86.8	75.7/86.2/77.0	86.6/93.0/76.3	—	—
PointMaze3v1Arena	74.1/90.5/88.7	67.4/75.5/81.7	69.2/73.4/78.3	65.3/77.4/80.3	65.8/80.8/82.2
PointMaze3v1Medium	73.3/82.3/85.1	61.3/75.7/76.3	72.3/74.9/72.0	71.1/79.6/77.9	73.1/76.2/76.4
PointMultiGoalArena	84.1/87.9/58.3	90.2/87.6/62.7	90.6/86.8/50.6	—	—
PointMultiGoalOrder	86.8/94.3/43.6	86.4/94.1/39.6	88.7/94.0/47.6	—	—
PointMultiGoalParallel	75.3/90.4/49.6	83.0/91.1/41.6	84.5/92.3/60.5	—	—
PointPushMultiGoal	92.3/94.7/59.9	89.3/92.9/61.8	91.8/94.0/57.1	—	—
PointPushRace	86.5/91.7/82.9	78.6/91.1/71.1	85.7/92.2/65.2	—	—
PointReachAvoidRace	60.6/78.3/82.1	62.9/77.9/77.6	59.9/78.1/81.0	—	—
PointReachRace	70.1/64.3/67.1	66.3/77.7/63.9	62.3/72.2/65.3	—	—
PointSoccerArena	91.0/94.9/68.9	90.2/96.7/75.4	90.4/97.2/74.6	90.0/95.9/75.8	—
PointSoccerIntervention	95.5/90.0/62.3	93.8/88.0/75.3	95.2/90.8/72.4	94.1/90.9/71.5	—
PointTrajectoryHI	99.6/21.1/7.8	99.6/22.3/10.8	99.6/23.0/7.6	—	—
Discrete					
CompetitiveButtonRace	97.8/63.7/20.1	98.7/82.7/24.8	86.5/80.1/22.5	—	—
KeyedHandoverProtocol	82.0/59.7/27.8	89.3/66.4/45.1	97.0/70.6/44.2	—	—
KeyedHandoverProtocol Competitive	94.2/70.3/40.9	95.7/79.6/44.8	96.6/57.9/43.4	—	—
KeyedHandoverProtocol MultiGoal	97.7/60.6/14.3	97.3/69.4/33.4	97.6/64.2/25.1	—	—
MixedCurling	84.7/85.7/34.6	84.9/88.0/40.7	88.6/87.3/42.1	—	—
Stack	93.4/84.5/49.5	95.9/72.1/42.7	85.3/85.3/63.5	—	—

A.21 HYPERPARAMETERS

Tables 21, 22, 23, and 24 list the hyperparameters of the continuous, goal-conditioned, discrete, and single-agent baselines.

A.22 OFFLINE DATASET GENERATION

Collection and storage. For offline data, we use the `H5` format and store synchronized agent observations, goals, actions, rewards, termination information, next-step fields, and the central state over complete physical episodes. The output observation is in the conventional dictionary format,

Table 21: **Hyperparameters of the continuous baselines.** ITD3+BC and ICQL are the Independent, local-objective adaptations of the recorded MATD3+BC and MACQL implementations. We use MACQL and MATD3+BC to denote their generic forms.

Parameter	MABC	MATD3+BC	MACQL	IICQ	OMIGA	InSPO	MABCQ
Shared backbone							
hidden_dim	256	256	256	256	256	256	256
lr_a	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0001
lr_c	—	0.0008	0.0008	0.0008	0.0008	0.0008	0.001
γ	—	0.98 | 0.99	0.98	0.98 | 0.99	0.98 | 0.99	0.98 | 0.99	0.99
τ	—	0.001 | 0.005	0.001	0.001 | 0.005	0.001 | 0.005	0.001 | 0.005	0.005
Method-specific							
MATD3+BC td3_bc_alpha 2.5; td3_bc_actor_warmup 0; policy_delay 1; policy_noise 0.05; noise_clip 0.5							
MACQL dataset_normalization fixed; cql_alpha 1; lse_temp 1; num_sampled_actions 10; cql_sample_noise_level 0.2; omar 1; omar_coe 1; omar_iters 2; omar_num_samples 20; omar_num_elites 5							
IICQ icq_alpha 1; icq_target_q_taken_beta 100							
OMIGA omiga_temperature 3; omiga_reg_coef 1; omiga_max_weight 20							
InSPO inspo_pretrain_steps 2000; inspo_pretrain_batch_size 256; inspo_alpha_start 1; inspo_alpha_end 0.001; inspo_t_max 100000 | 50000; inspo_is_ratio_min 0 | 1; inspo_is_ratio_max 1 | 100; inspo_cql_coeff 0							
MABCQ latent_dim 10; candidates 10; optimism 0.8; phi 0.05; twin_q_min_weight 0.75; policy_delay 1; pretrain_steps 2000; vae_lr 0.0001							

Table 22: **Goal-conditioned hyperparameters.** Frozen state-based configurations; — marks a parameter the method does not use. Future goals are drawn uniformly from later steps of the same episode (next_achieved_goal); 0.8 is the probability of attempting a relabel, which the validity checks of the task may reject, and accepted goals use the goal, reward, and termination definitions of the task. *PointPushMultiGoal uses a recorded object-relative transform.

Parameter	GCBC	GCIQL	GCIVL
Training and evaluation			
Updates / batch size		50k / 512	
Training seeds		42, 62, 82, 102, 122	
Validation		10 episodes every 10k updates	
Held-out evaluation		100 episodes, mean action	
Goal relabeling probability		0.8 (one candidate per proposal)	
Network and optimization			
Hidden layers / activation		256×2 / ReLU	
Optimizer / actor learning rate		Adam / 0.0003	
Policy std. (training likelihood)		1 (fixed)	
Input normalization / transform		fitted once on the dataset / identity*	
Per algorithm			
Q-function learning rate	—	0.0003	—
Value-function learning rate	—	0.0003	0.0003
Discount γ	—	0.99	0.99
Target-update rate τ	—	0.005	0.005
Expectile	—	0.7	0.7
Actor weight	unweighted	$\min\{e^{3A}, 100\}$	$\min\{e^{3A}, 100\}$

Table 23: **Discrete-action hyperparameters.** Categorical implementations only; — marks a parameter the method does not use. All methods sample aligned joint transitions without goal relabeling and use no recurrent network. MAICQ is the feed-forward one-step variant ($\lambda = 0$), trains the critic before the actor in each update, and requires a shared reward and termination, so it is reported only on Stack, KeyedHandoverProtocol, and KeyedHandoverProtocolMultiGoal.

Parameter	MABC	MAIQL	MAICQ
Training and evaluation			
Transitions / updates / batch size		50k / 50k / 512	
Training seeds		42, 62, 82, 102, 122	
Held-out evaluation		100 episodes, greedy	
Validation interval (updates)		10k	
Validation episodes	10	10	20
Network and optimization			
Hidden layers / activation		64×2 / ReLU	
Training paradigm	Independent	Independent	CTDE
Critic structure	—	twin categorical Q, expectile V	nonnegative linear mixer (width 32)
Optimizer	Adam	Adam	RMSprop ($\alpha = 0.99$, $\epsilon = 10^{-5}$)
Actor / critic learning rate	0.0003 / —	0.0003 / 0.0003	0.0005 / 0.0001
Value-function learning rate	—	0.0003	—
Discount γ	—	0.99	0.99
Target update	—	soft, $\tau = 0.005$	hard copy every 600
Expectile	—	0.7	—
Actor weight	unweighted	$\min\{e^{3A}, 100\}$	batch-normalized softmax
Temperatures β_π / β_Q	—	—	0.5 / 1000
Gradient-norm clip	—	—	10

Table 24: **Single-offline hyperparameters.** All three algorithms share the vector encoder: two 256-unit hidden layers with ReLU, no batch or layer normalization, no dropout, and no separate last-layer activation.

Parameter	BCQ	CQL	TD3+BC
Shared			
`batch_size`		512	
γ		0.99	
`compile_graph`		False	
τ		0.005	
`n_critics`		2	
Per algorithm			
`actor_learning_rate`	0.001	0.0001	0.0003
`critic_learning_rate`	0.001	0.0003	0.0003
`update_actor_interval`	1	—	2
`n_action_samples`	100	10	—
BCQ `imitator_learning_rate` 0.001; `lam` 0.75; `action_flexibility` 0.05; `rl_start_step` 0; `beta` 0.5			
CQL `temp_learning_rate` 0.0001; `alpha_learning_rate` 0.0001; `initial_temperature` 1; `initial_alpha` 1; `alpha_threshold` 10; `conservative_weight` 5; `soft_q_backup` False; `max_q_backup` False			
TD3+BC `target_smoothing_sigma` 0.2; `target_smoothing_clip` 0.5; `alpha` 2.5			

similar to the Fetch environments (Plappert et al., 2018). Each transition is one synchronized step of all agents. Expert and Suboptimal sources are online MASAC checkpoints (Section 4.1) selected through deterministic evaluation on the task metric. Expert checkpoints score in $[0.80, 1]$, while Suboptimal checkpoints score in $[0.30, 0.60]$. Competitive self-play sources use analogous thresholds based on win frequency. For every source, we collect complete episodes without filtering by success or return until the transition budget is reached (Appendices A.6 and A.17), truncate the last episode at the budget, and keep the true next observation at the truncation point. The `done` flag records only environment termination, so time-outs and budget truncations remain bootstrappable, and sequence sampling never crosses an episode boundary or a truncation point.

Mixed and Stitch. A *Mixed* pool draws exactly half of its transitions from each parent, *Guided* and *Random*. Parent episodes are shuffled, and the last one is truncated at the quota, so unequal episode lengths change the number of episodes per parent but not the 50/50 transition ratio. In a *Stitch* pool, all agents switch controllers simultaneously after ten steps in the continuous tasks and after two steps in Discrete Stack. The incoming controller continues from the current state and inherits no internal state from the outgoing one. Episodes that terminate before the switch are retained, and each pool contains at least one episode that reaches both phases. We report the realized share of each source as the fraction of retained transitions collected in its phase.

Role composition. In a *Combination* pool (Figure 11), each role executes the corresponding actor of its assigned checkpoint without retraining. For *SA-Extended*, each role is trained separately with SAC for 100k steps while its partners follow the task's Guided controller (PPO with the same budget for Discrete Stack). Every 5k steps, we evaluate the role on 20 deterministic episodes and keep the checkpoint with the highest local success, preferring the earlier one on ties. The selected local policies then collect new joint episodes under the original task rewards.

Action perturbations. Learned sources (*PretrainedExpert*, *PretrainedSuboptimal*, and *Combination*) replace the joint action with a uniformly sampled valid action with probability 0.1 per step, jointly for all agents. The scripted *Guided* controller instead adds uniform noise of 10% of each coordinate's half-range and clips the result, or, for discrete actions, replaces each agent's action

Table 25: **Tasks in the Relative, Partial, and Features comparison.** All 16 configurations enter Figure 6 (A/B) on matched Guided and Mixed physical records. The last column marks inclusion in the separate 14-task Relative–Partial response summary. Registry prefixes and version suffixes are omitted.

Platform	Task configuration	14-task response
PyBullet	Asyn dual	Yes
	Connect dual	Yes
	Curling dual	Yes
	Handover dual	Yes
	Handover triple	Yes
	HandoverActiveSuction dual	No
	HandoverIntervention dual	Yes
	HandoverPush dual	Yes
	HandoverPush triple	Yes
	HandoverPush quad	Yes
	Insert dual	Yes
	InsertActiveSuction dual	No
	InsertIntervention dual	Yes
	PushButton dual	Yes
MuJoCo	PandaHandover	Yes
	PandaValveClose	Yes
Total	**16 configurations**	**14**

independently with probability 0.1. *Mixed* inherits the noise of its parents, *Stitch* keeps the Guided-phase noise without additional replacement, and *SA-Extended* is collected without noise.

A.23 Observation Interfaces and Learning Protocols

The Obs. column in Table 28 lists each actor's observation. The desired goal is appended to it, while the achieved goal is stored separately. The central state is used for centralized training and analysis but is not available to decentralized actors. Observation modes and image features do not count as separate task configurations.

Relative (R). The 35 PyBullet and nine MuJoCo Panda configurations use task-specific local geometric features. In PyBullet, seven joint angles are prepended and registered positions are expressed in the end-effector frame as $R_i^{\mathsf{T}}(p - p_{\mathrm{ee},i})$, with R_i the end-effector-to-world rotation. MuJoCo Panda tasks use world-axis differences $p - p_{\mathrm{ee},i}$ together with joint and finger positions and velocities, end-effector position and velocity, and control-target errors. Object, partner, contact, and stage fields follow each task's schema, so dimensions differ across tasks.

Partial (R+P). R+P marks tasks that also offer a Partial variant as an alternative to Relative. The two modes are not concatenated. Partial retains the Relative frame but zeros every field that depends on an entity outside the agent's visibility region, including derived fields, history fields, and dynamic goals. It appends visibility indicators while keeping proprioception and public desired targets available. Table 25 lists the 16 configurations used in the observation comparison, all of which appear in Figure 6A/B. The two active-suction supplements are excluded only from the Relative–Partial response summary, which covers the remaining 14 tasks.

Features. Built on the Partial tasks, Features concatenates seven joint angles, three end-effector/base coordinates, a 128-dimensional image embedding, and the desired goal, rather than using images alone. Images follow each agent's visibility, are resized to 84×84, and are scaled to $[0, 1]$ without augmentation. The encoder, a three-stage IMPALA CNN with 16/32/32 channels, is trained by behavior cloning on the matching task and source pool for 5k updates (batch 128, Adam, learning rate 3×10^{-4}), then frozen and shared across algorithms, seeds, and evaluation. Because

Table 26: **Task configurations by interaction family and platform.**

Family	PyBullet	MuJoCo	Discrete	Total
Cooperation	11	8	1	20
Competition	4	5	2	11
Multi-Goal	4	5	1	10
Transition	7	3	1	11
Hybrid	1	1	1	3
Trajectory-Tracking	4	1	0	5
Intervention	4	1	0	5
All configurations	**35**	**24**	**6**	**65**

FEATURES runs use 100k updates, they are kept apart from the matched-budget RELATIVE-PARTIAL comparison.

NATIVE (N). The 13 MuJoCo Point tasks, the two DualGripper diagnostics, and the six discrete tasks use environment-defined actor schemas without the RELATIVE/PARTIAL transformation. Point tasks expose per-agent observation slices and local goals, DualGripper actors receive local features that already contain relative positions and goals, and discrete actors receive task-specific symbolic or private protocol fields. NATIVE refers to the origin of the schema. It does not imply world-frame coordinates, full observability, or access to the central state. These tasks do not enter the RELATIVE–PARTIAL comparison.

Goal relabeling. Each of the L transitions in a retained episode has an equal probability of being selected as t. We draw j uniformly from $\{t, \ldots, L-1\}$ and, with probability 0.8, propose the achieved goal at the stored next state of transition j. A single candidate is tested using the same future index for all agents. The candidate is either accepted for every agent or rejected for all of them, in which case the original row is retained. Candidates are rejected when the state is invalid or in collision, or when the new goal would alter the completion of an earlier stage. Relabeling the final stage requires the source and future transitions to share that stage, and coupled tasks preserve the recorded grasp, transfer, release, and transport events: Handover, for instance, changes the payload target while keeping the handover history, and PointMultiGoalArena changes the final waypoints while keeping earlier completion latches. For an accepted goal G' and recorded history h_t, we recompute

$$r_t^{i\prime} = R^i_{\text{task}}(s_t, a_t, s_{t+1}, h_t, G'), \qquad m_t^{i\prime} = 1 - T^i_{\text{task}}(s_t, s_{t+1}, h_t, G'),$$

together with the dependent local inputs. Rejected candidates keep the original reward and terminal mask, so the realized relabeling rate can fall below 0.8.

A.24 TASK CATALOG

Table 27 states the rule and design purpose of each family, and Table 26 counts the configurations by family and platform; the three active-suction variants are counted in the families of their base tasks. Table 28 lists each registered task once with its family, agent count, horizon, and per-agent interface dimensions. PyBullet box dimensions are full lengths in meters, whereas MuJoCo box sizes are generally half-extents; Point scene units and discrete pixels are not meters.

Table 27: **Rule-defined interaction families.**

Family	Defining rule	Example	What it isolates
Cooperation	Agents jointly satisfy one completion predicate	Lift, Connect	Simultaneous contribution; no single agent's subgoal suffices
Competition	Individual or team objectives conflict	CompetitionPush, PointReachRace	Strategic interference, and outcomes that depend on the side
Multi-Goal	A goal set, order or switch governs completion	PushSequential	Completion across several targets rather than one terminal goal
Transition	One role creates the state a later role acts on	Handover	Ordered role dependence and the transfer of control
Hybrid	Shared progress coexists with individual payoff	Curling, PointSoccerArena	Collective completion pulling against individual incentive
Trajectory-Tracking	An ordered path or a repeated pattern is required	HILettersRepeat	Sustained execution, with first completion kept apart from repetitions
Intervention	An exogenous event fires during execution	PushIntervention	Continuation after a dynamics change, judged by the task-native rule

Table 28: **Registered task interfaces.** All 65 configurations, grouped by platform. N is the agent count and H the horizon in steps. Obs. is the observation interface: R RELATIVE, R+P RELATIVE with a PARTIAL variant, N native. O/A gives per-agent actor-observation and action dimensions, excluding the joint mode, with D marking a discrete action space.

Task	Family	N	H	Obs.	O/A
PyBullet					
Asyn dual	Transition	2	80	R+P	30/3
CompetitionButton dual	Competition	2	50	R	55/3
CompetitionPush dual	Competition	2	50	R	32/3
Connect dual	Cooperation	2	50	R+P	50/3
Curling dual	Hybrid	2	80	R+P	51/3
GOODLettersRepeat quad	Trajectory-Tracking	4	1,000	R	16,19/3[†]
HILetters dual hiMulti	Trajectory-Tracking	2	500	R	16/3
HILettersRepeat dual hiMulti	Trajectory-Tracking	2	1,000	R	16/3
Handover dual	Transition	2	80	R+P	23/3
Handover triple	Transition	3	250	R+P	29/3
HandoverActiveSuction dual	Transition[‡]	2	80	R+P	23/4
HandoverIntervention dual	Intervention	2	80	R+P	23/3
HandoverPush dual	Transition	2	60	R+P	23/3
HandoverPush quad	Transition	4	80	R+P	32/3
HandoverPush triple	Transition	3	80	R+P	29/3
Insert dual	Cooperation	2	80	R+P	50/3
InsertActiveSuction dual	Cooperation[‡]	2	80	R+P	50/4
InsertIntervention dual	Intervention	2	80	R+P	50/3
Lift dual	Cooperation	2	50	R	29/3
Lift quad	Cooperation	4	50	R	41/3
Lift triple	Cooperation	3	50	R	35/3
LiftActiveSuction dual	Cooperation[‡]	2	50	R	29/4

continued on next page

Table 28 continued

Task	Family	N	H	Obs.	O/A
LiftIntervention dual	Intervention	2	50	R	29/3
Push dual	Cooperation	2	50	R	30/3
PushButton dual	Multi-Goal	2	50	R+P	46/3
PushIntervention dual	Intervention	2	50	R	30/3
PushSequential dual	Multi-Goal	2	50	R	29/3
Reach dual	Cooperation	2	50	R	16/3
ReachAvoid dual	Cooperation	2	60	R	42/3
ReachSequential dual	Multi-Goal	2	50	R	19/3
Soccer dual	Competition	2	80	R	37/3
Stack dual	Cooperation	2	50	R	36/3
TicTacToe dual	Competition	2	100	R	97/3
TrajectoryMPG triple	Trajectory-Tracking	3	800	R	16,19/3[†]
WindowsMultiGoal dual	Multi-Goal	2	120	R	51/3
MuJoCo					
DualGripperHandoverProxy	Transition	2	160	N	44/8
DualGripperValveTurnCloseProxy	Cooperation	2	160	N	44/8
PandaCableArc	Cooperation	2	650	R	262/4
PandaCablePlace	Cooperation	2	600	R	262/4
PandaCableSShape	Cooperation	2	700	R	262/4
PandaCableUShape	Cooperation	2	700	R	262/4
PandaDualLift	Cooperation	2	400	R	70/4
PandaHandover	Transition	2	700	R+P	107/4
PandaSlideHandover	Transition	2	800	R	77/4
PandaStack	Cooperation	2	800	R	127/4
PandaValveClose	Cooperation	2	500	R+P	68/4
PointHeterogeneousMultiGoal	Multi-Goal	2	450	N	18/2
PointMaze3v1Arena	Competition	4	500	N	12/2
PointMaze3v1Medium	Competition	4	500	N	12/2
PointMultiGoalArena	Multi-Goal	2	500	N	11/2
PointMultiGoalOrder	Multi-Goal	2	450	N	18/2
PointMultiGoalParallel	Multi-Goal	2	450	N	18/2
PointPushMultiGoal	Multi-Goal	2	450	N	18/2
PointPushRace	Competition	2	300	N	18/2
PointReachAvoidRace	Competition	2	300	N	18/2
PointReachRace	Competition	2	300	N	18/2
PointSoccerArena	Hybrid	3	500	N	17/2
PointSoccerIntervention	Intervention	3	500	N	19/2
PointTrajectoryHI	Trajectory-Tracking	2	400	N	14/2
Discrete					
CompetitiveButtonRace	Competition	2	24	N	39/D7
KeyedHandoverProtocol	Transition	2	18	N	40/D10
KeyedHandoverProtocolCompetitive	Competition	2	18	N	40/D10
KeyedHandoverProtocolMultiGoal	Multi-Goal	2	24	N	40/D10
MixedCurling	Hybrid	2	32	N	39/D13
Stack	Cooperation	2	28	N	39/D11

[†]Observation dimensions vary by seat: $16/19/19/16$ for GOODLettersRepeat and $16/19/16$ for TrajectoryMPG. [‡]Active-suction action-space variant, counted in the family of its base task.

A.25 Visual Task Sequences

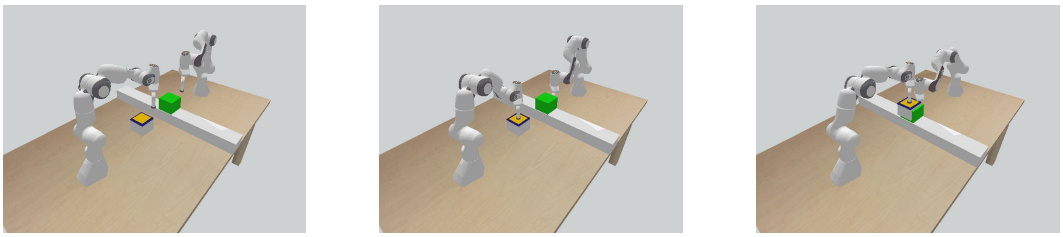
For each registered configuration, the three frames show an initial scene, an intermediate stage, and a terminal scene from left to right, and the text states the completion criterion.

PyBullet

PyBullet Asyn dual

Completion: The moved object reaches its dynamic goal relative to the other object.

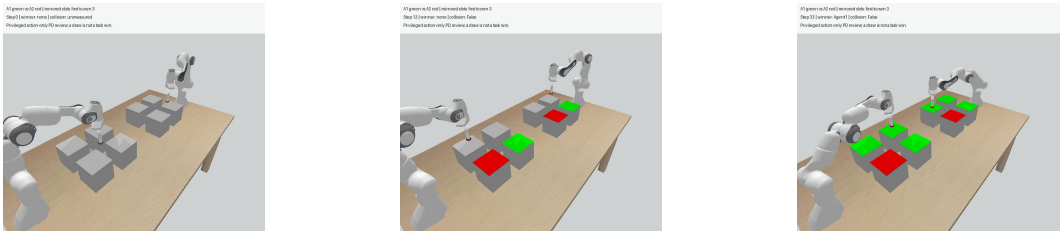
Transition; 2 agents; horizon 80



PyBullet CompetitionButton dual

Completion: Each arm is judged against its own button goal; winner and draw are retained.

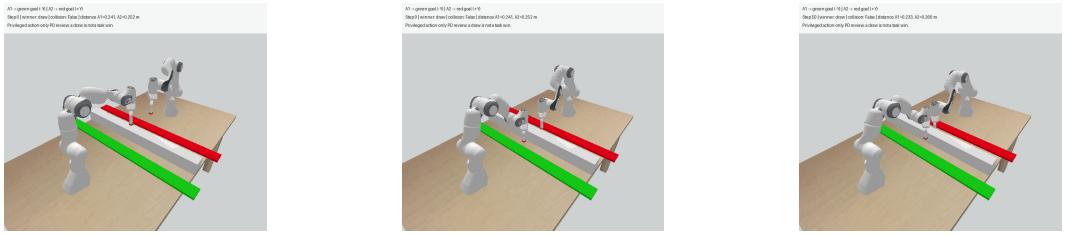
Competition; 2 agents; horizon 50



PyBullet CompetitionPush dual

Completion: Each arm pushes toward its own opposed goal; winner, draw, and per-agent outcomes are recorded.

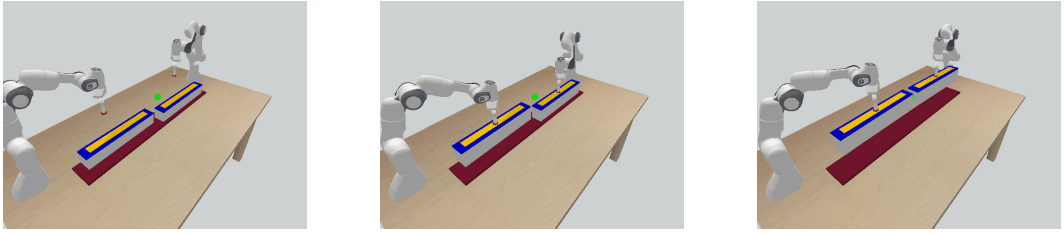
Competition; 2 agents; horizon 50



PyBullet Connect dual

Completion: Both connector pieces reach their prescribed matching positions.

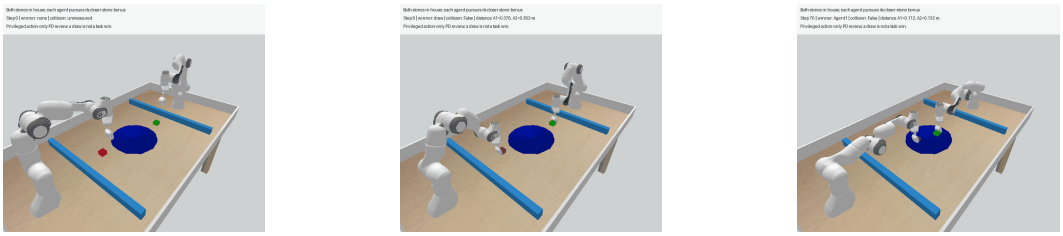
Cooperation; 2 agents; horizon 50



PyBullet Curling dual

Completion: Common completion and each agent's target distance and return are recorded separately.

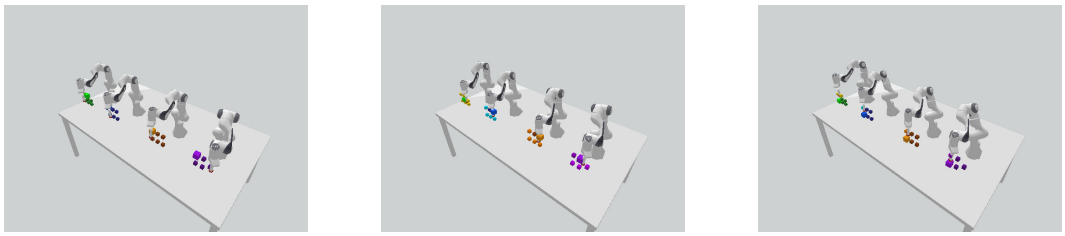
Hybrid; 2 agents; horizon 80



PyBullet GOODLettersRepeat quad

Completion: Complete G-O-O-D patterns are counted within the fixed horizon.

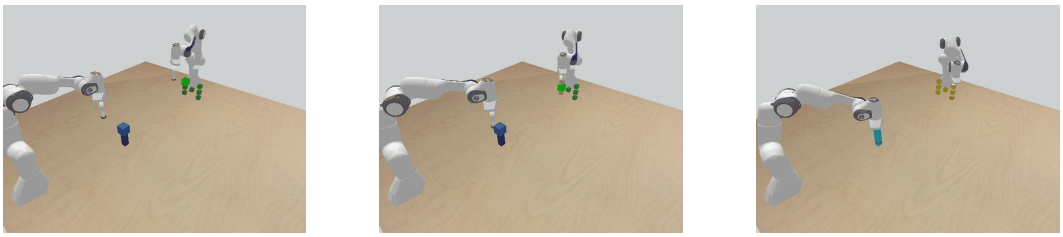
Trajectory-Tracking; 4 agents; horizon 1000



PyBullet HILetters dual hiMulti

Completion: Both arms finish their H and I letter paths; this is a one-shot success outcome.

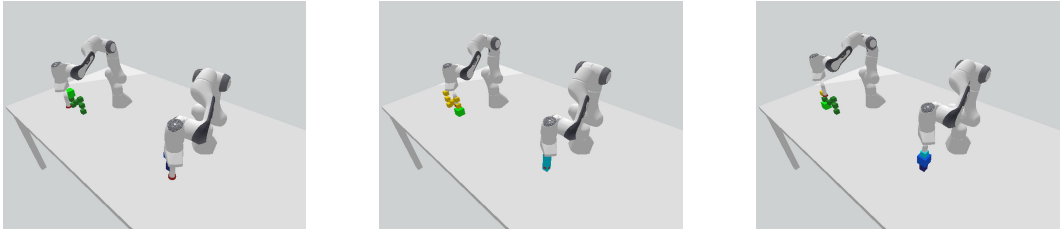
Trajectory-Tracking; 2 agents; horizon 500



PyBullet HILettersRepeat dual hiMulti

Completion: Complete H-I patterns are counted within the fixed horizon.

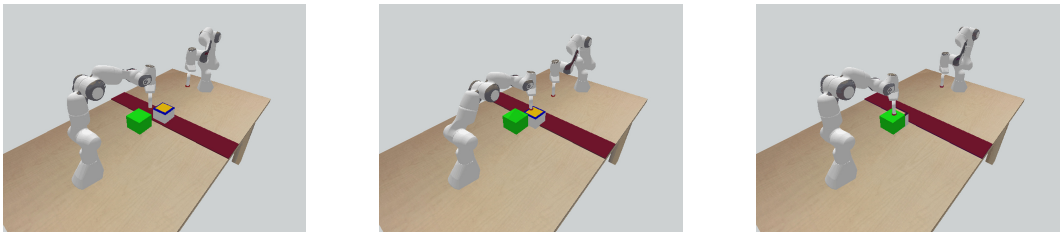
Trajectory-Tracking; 2 agents; horizon 1000



PyBullet Handover dual

Completion: The shared object reaches the final target after transfer between arms.

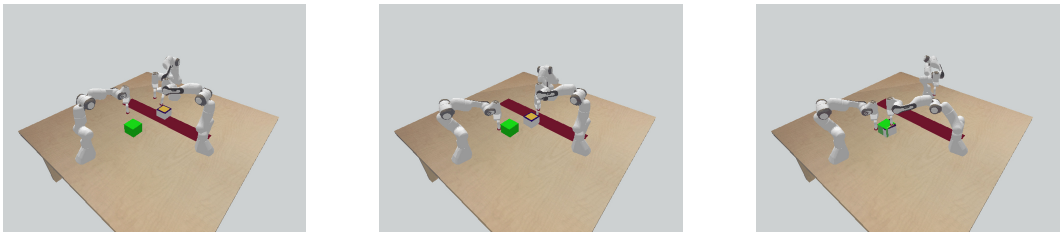
Transition; 2 agents; horizon 80



PyBullet Handover triple

Completion: The shared object reaches the final target after the arm-to-arm handover.

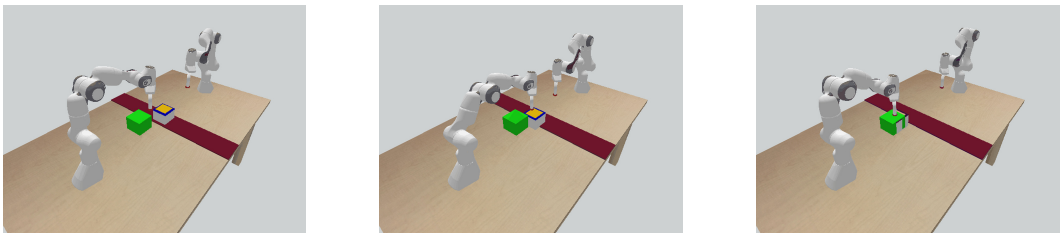
Transition; 3 agents; horizon 250



PyBullet HandoverActiveSuction dual

Completion: The shared object reaches the final target after transfer.

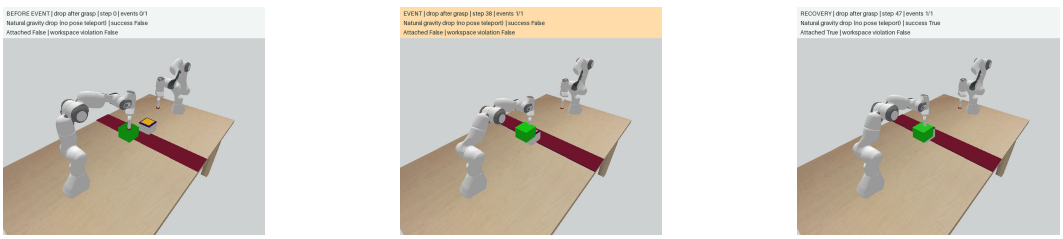
Transition (active-suction variant); 2 agents; horizon 80



PyBullet HandoverIntervention dual

Completion: The shared object must reach the final target after intervention; the event alone is not success.

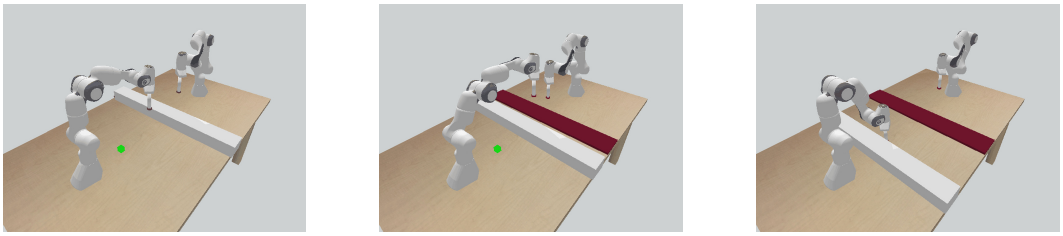
Intervention; 2 agents; horizon 80



PyBullet HandoverPush dual

Completion: The object reaches the final goal after the push-and-relay sequence.

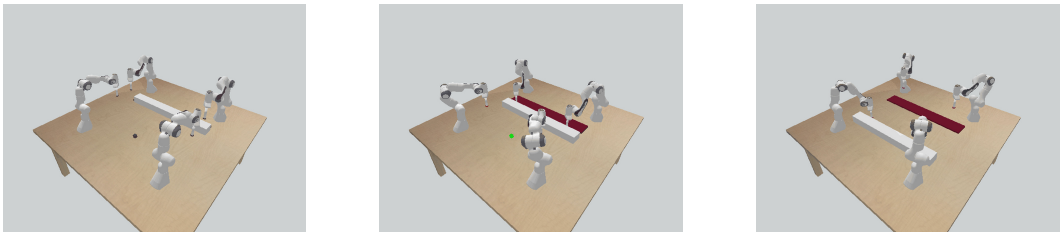
Transition; 2 agents; horizon 60



PyBullet HandoverPush quad

Completion: The object is relayed through the arms and reaches the final target region.

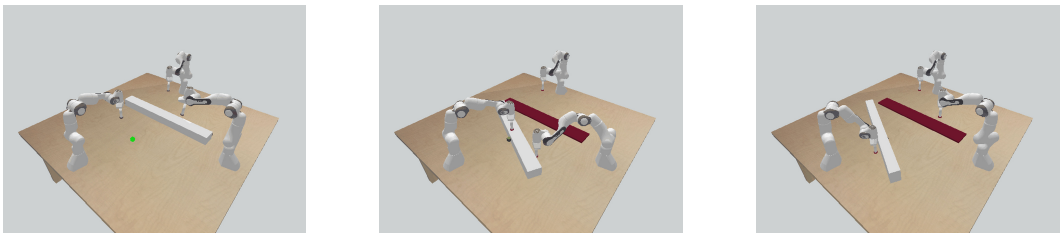
Transition; 4 agents; horizon 80



PyBullet HandoverPush triple

Completion: The object is relayed through the arms and reaches the final target region.

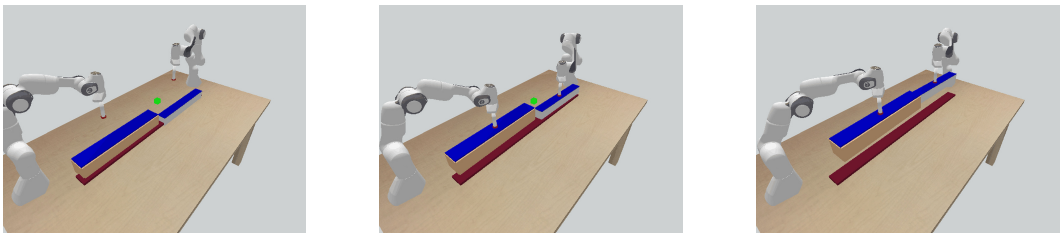
Transition; 3 agents; horizon 80



PyBullet Insert dual

Completion: Both component-to-goal distances meet their thresholds, including the insertion relation.

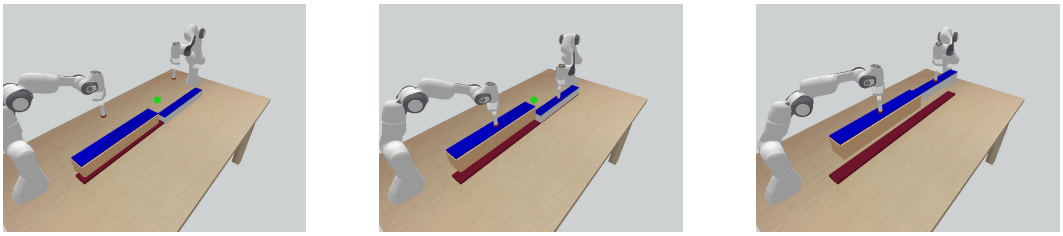
Cooperation; 2 agents; horizon 80



PyBullet InsertActiveSuction dual

Completion: Both component-to-goal distances meet the insertion thresholds.

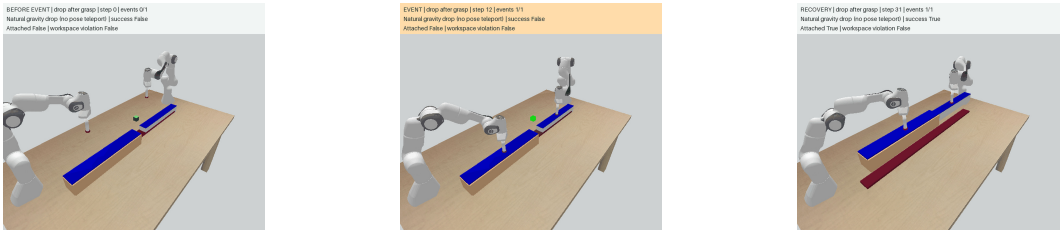
Cooperation (active-suction variant); 2 agents; horizon 80



PyBullet InsertIntervention dual

Completion: Both insertion goals must be met after intervention; the event alone is not success.

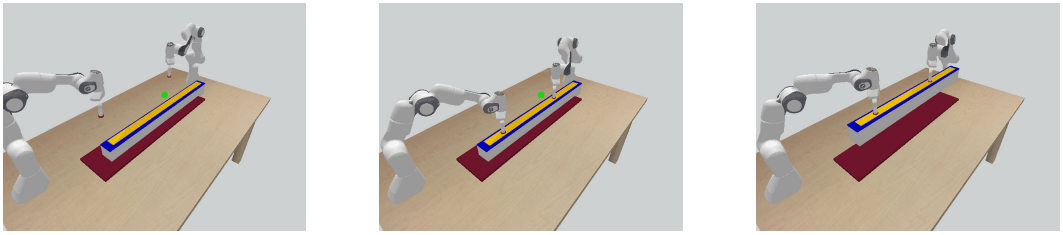
Intervention; 2 agents; horizon 80



PyBullet Lift dual

Completion: The jointly held object reaches its target with the dual-arm tilt check satisfied.

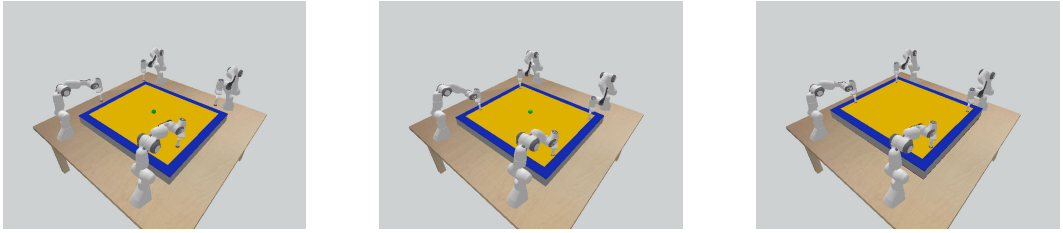
Cooperation; 2 agents; horizon 50



PyBullet Lift quad

Completion: The jointly supported object reaches its target; one arm arriving alone is insufficient.

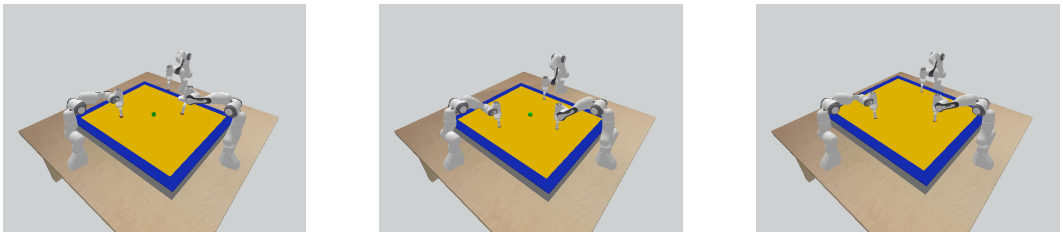
Cooperation; 4 agents; horizon 50



PyBullet Lift triple

Completion: The jointly supported object reaches its target; one arm arriving alone is insufficient.

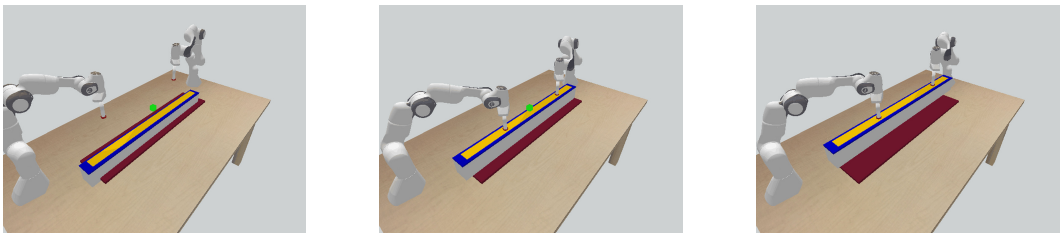
Cooperation; 3 agents; horizon 50



PyBullet LiftActiveSuction dual

Completion: The jointly lifted object reaches its goal with the original dual-arm tilt check.

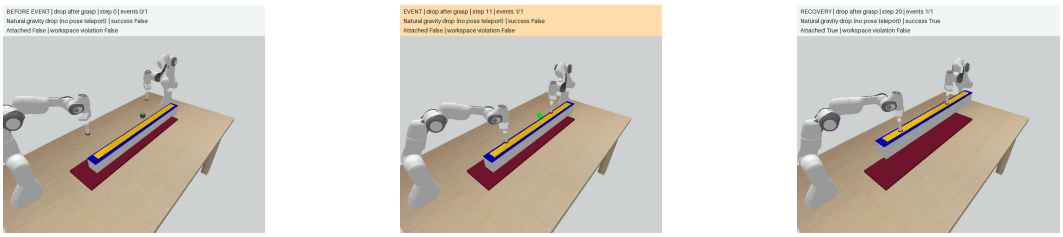
Cooperation (active-suction variant); 2 agents; horizon 50



PyBullet LiftIntervention dual

Completion: The original lift goal and tilt check still determine success after intervention.

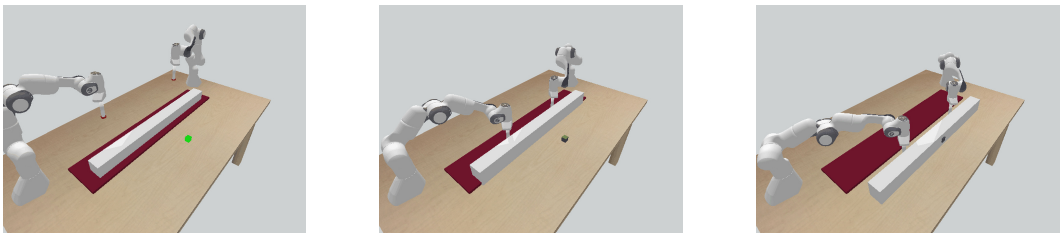
Intervention; 2 agents; horizon 50



PyBullet Push dual

Completion: World-X error is at most 5 cm and wrapped yaw error at most 20 degrees; Y/Z are contextual.

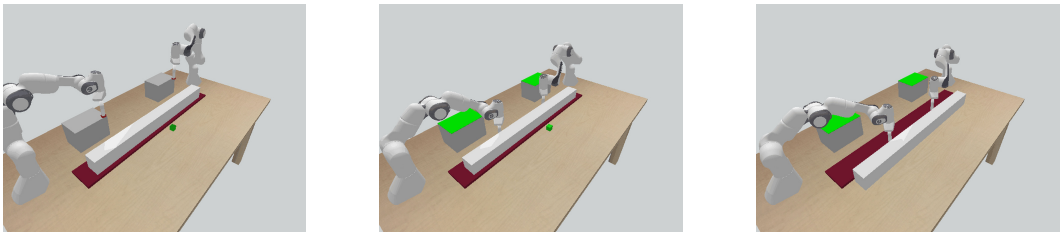
Cooperation; 2 agents; horizon 50



PyBullet PushButton dual

Completion: The button prerequisite is met before the object reaches its goal.

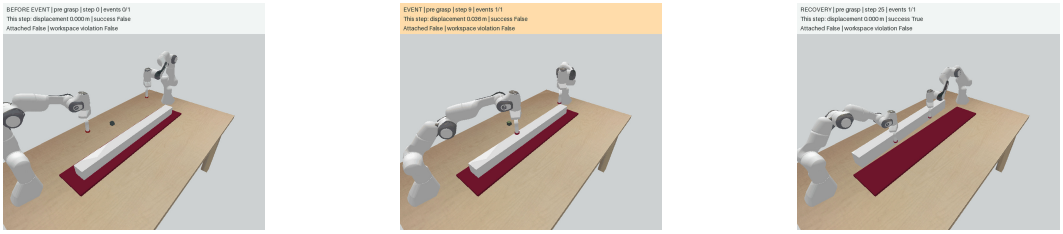
Multi-Goal; 2 agents; horizon 50



PyBullet PushIntervention dual

Completion: The original X/yaw push criterion still determines success after intervention.

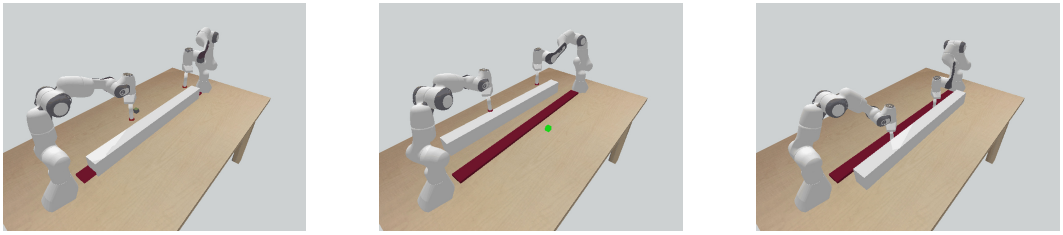
Intervention; 2 agents; horizon 50



PyBullet PushSequential dual

Completion: The shared object completes both ordered push goals.

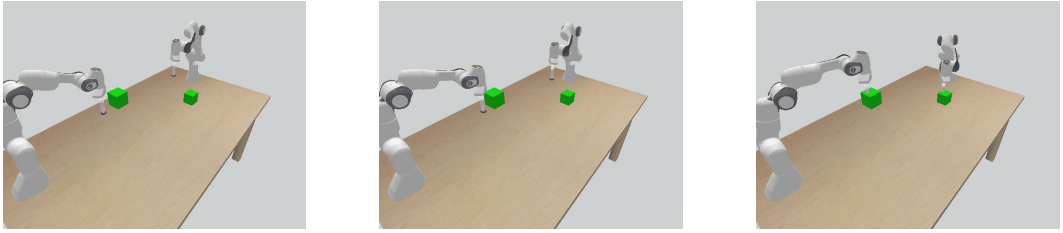
Multi-Goal; 2 agents; horizon 50



PyBullet Reach dual

Completion: Both end effectors simultaneously enter their respective goal regions.

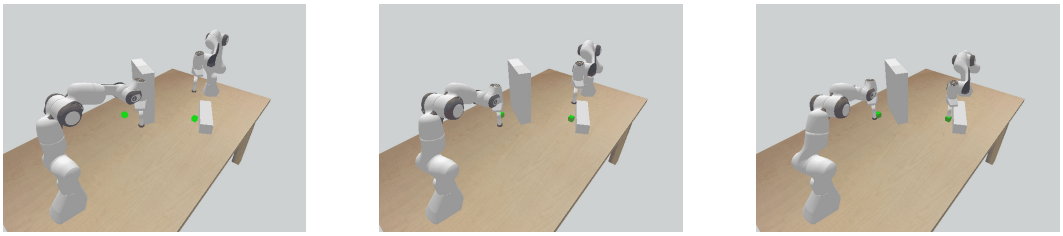
Cooperation; 2 agents; horizon 50



PyBullet ReachAvoid dual

Completion: Both arms reach their goals without termination by the safety-collision rule.

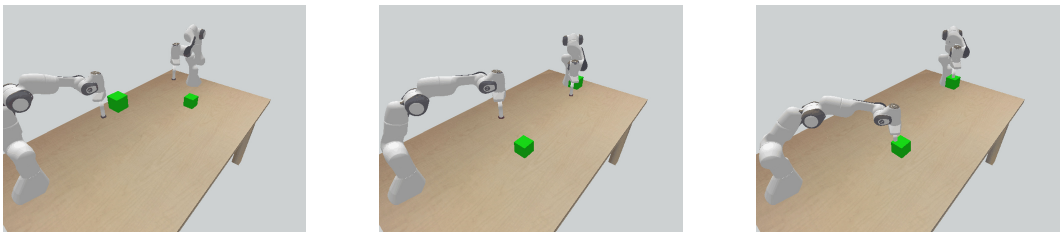
Cooperation; 2 agents; horizon 60



PyBullet ReachSequential dual

Completion: The second goal stage is completed; reaching the first stage alone is not success.

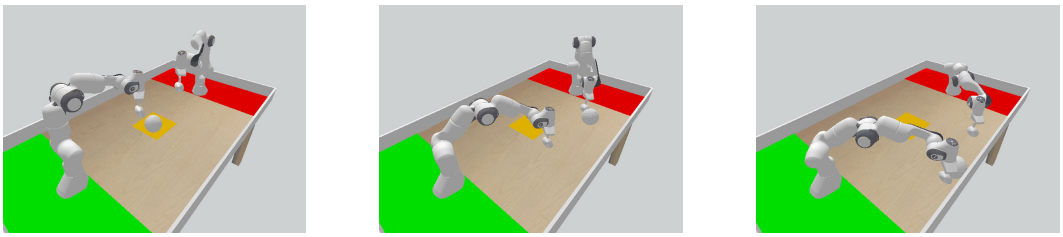
Multi-Goal; 2 agents; horizon 50



PyBullet Soccer dual

Completion: A goal in the corresponding scoring region determines each agent's outcome and winner.

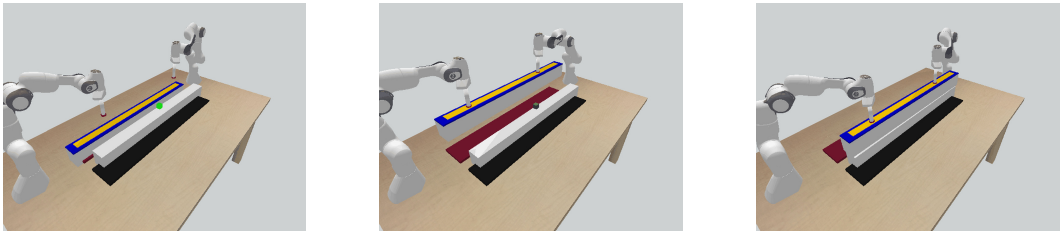
Competition; 2 agents; horizon 80



PyBullet Stack dual

Completion: The moved object's center is within 5 cm of the stack goal defined relative to the base.

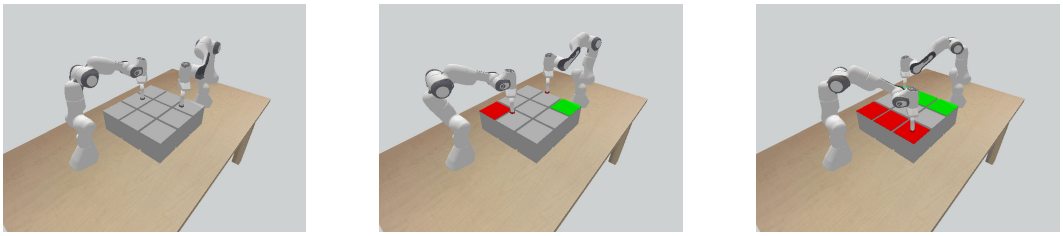
Cooperation; 2 agents; horizon 50



PyBullet TicTacToe dual

Completion: The board reaches a winning or terminal configuration; winner and draw are retained.

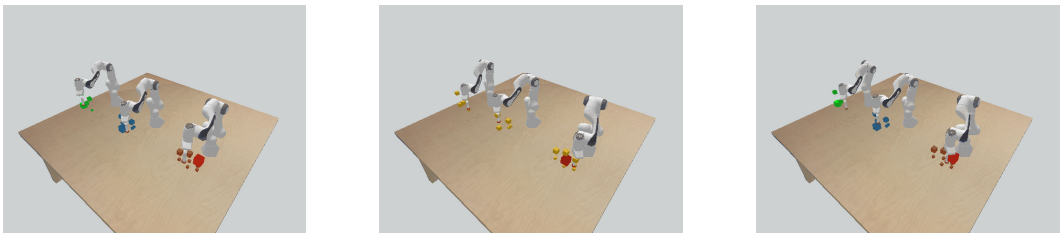
Competition; 2 agents; horizon 100



PyBullet TrajectoryMPG triple

Completion: All three end effectors traverse their ordered M, P, and G waypoints; completed patterns are counted.

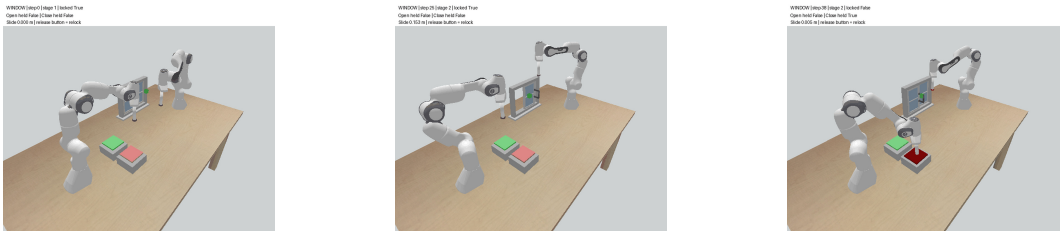
Trajectory-Tracking; 3 agents; horizon 800



PyBullet WindowsMultiGoal dual

Completion: The window first opens beyond 0.150 m, then closes below 0.020 m while the corresponding button is held.

Multi-Goal; 2 agents; horizon 120

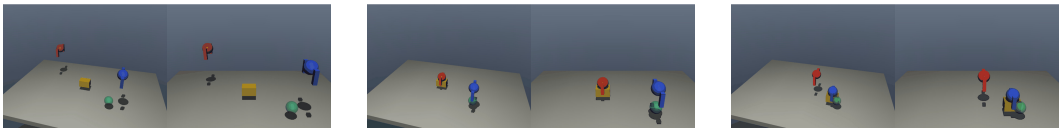


MuJoCo

MuJoCo DualGripperHandoverProxy

Completion: The proxy's target, stage, and holding-state predicates are satisfied.

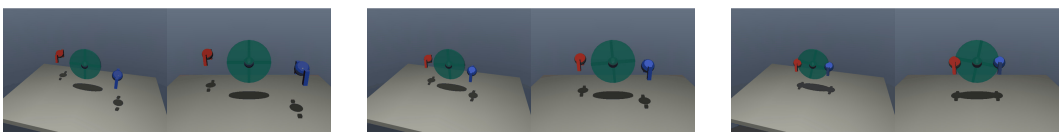
Transition; 2 agents; horizon 160



MuJoCo DualGripperValveTurnCloseProxy

Completion: The proxy's target, stage, and holding-state predicates are satisfied.

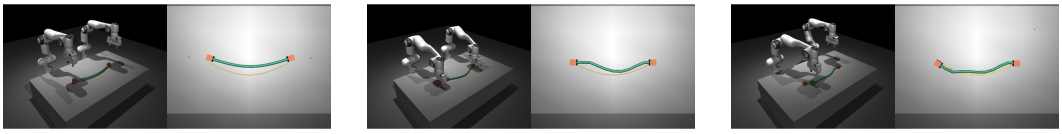
Cooperation; 2 agents; horizon 160



MuJoCo PandaCableArc

Completion: The whole cable meets the arc-shape target and remains stable after release.

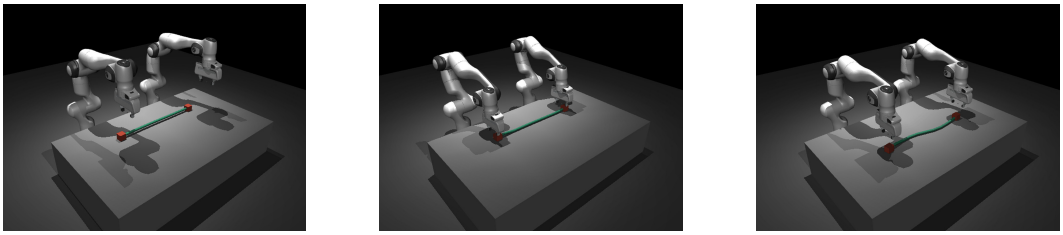
Cooperation; 2 agents; horizon 650



MuJoCo PandaCablePlace

Completion: Transport history, cable keypoint targets, release, and stability requirements are met.

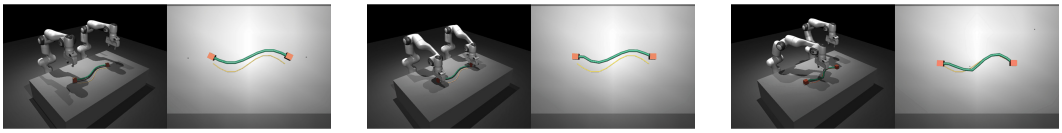
Cooperation; 2 agents; horizon 600



MuJoCo PandaCableSShape

Completion: The cable meets the S-shape keypoint target and remains stable after release.

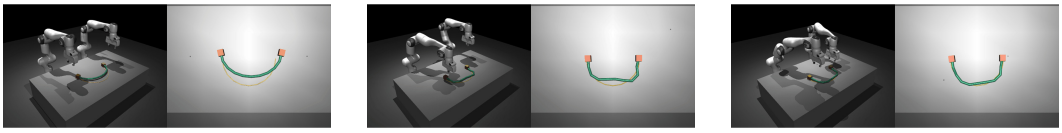
Cooperation; 2 agents; horizon 700



MuJoCo PandaCableUShape

Completion: The cable meets the U-shape keypoint target and remains stable after release.

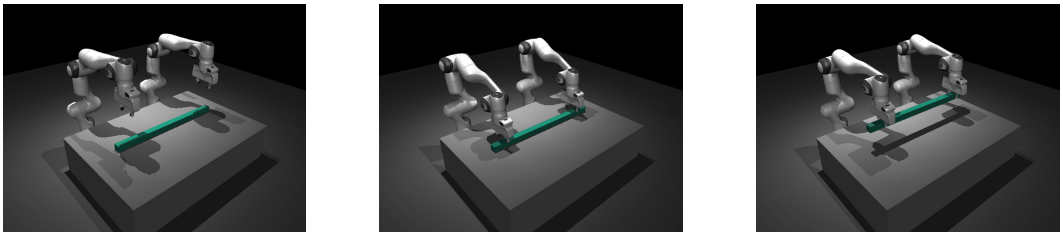
Cooperation; 2 agents; horizon 700



MuJoCo PandaDualLift

Completion: The mean grasp-point center is within 2.5 cm of goal while both grippers maintain two-finger contact.

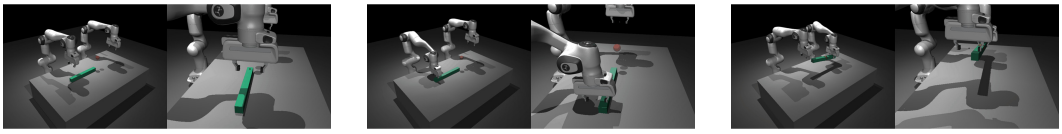
Cooperation; 2 agents; horizon 400



MuJoCo PandaHandover

Completion: The receiver transports and supports the object at goal after a genuine handover and sender release.

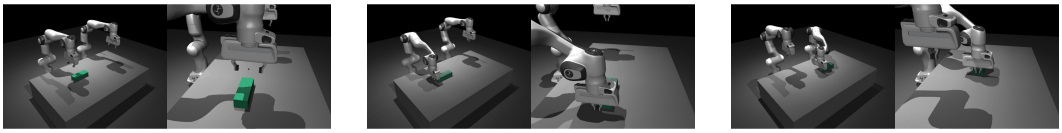
Transition; 2 agents; horizon 700



MuJoCo PandaSlideHandover

Completion: Each arm supports and pushes at least 5.5 cm; object-center error is below 1.8 cm without leaving the table.

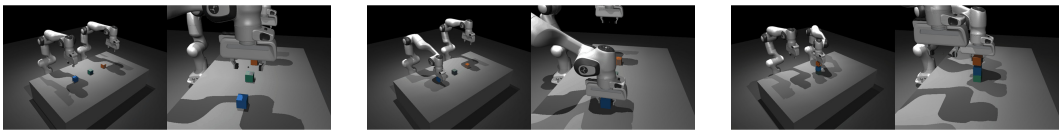
Transition; 2 agents; horizon 800



MuJoCo PandaStack

Completion: The centers of all three blocks lie within 1.8 cm of their respective stack goals.

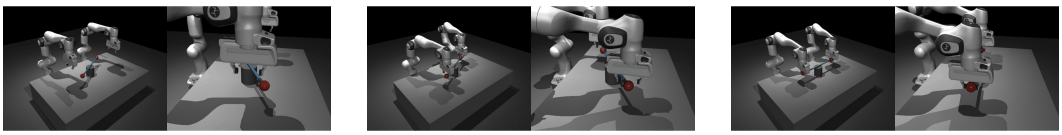
Cooperation; 2 agents; horizon 800



MuJoCo PandaValveClose

Completion: Valve angle error is below 0.06 rad and both endpoint errors are below 1.8 cm while both grippers hold.

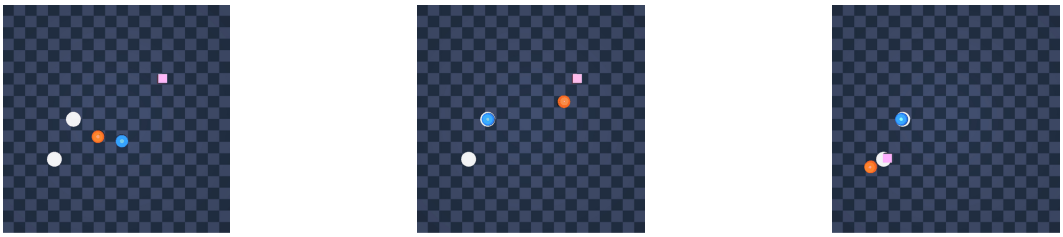
Cooperation; 2 agents; horizon 500



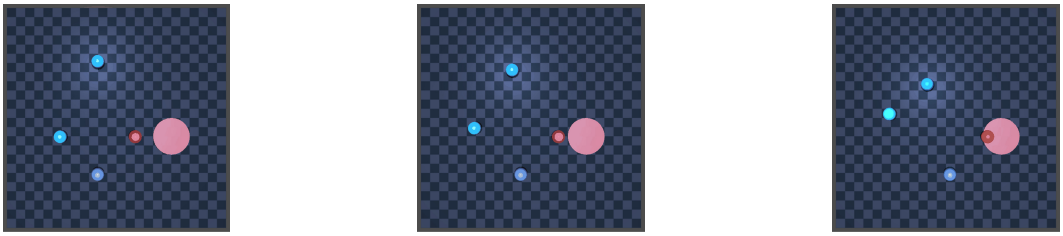
MuJoCo PointHeterogeneousMultiGoal

Completion: One agent reaches its navigation goal while the other pushes its object to goal.

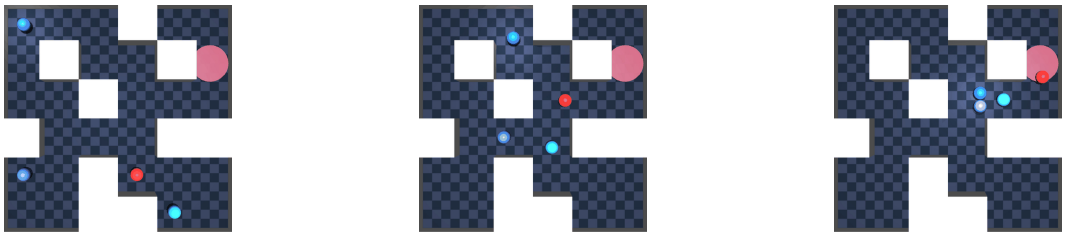
Multi-Goal; 2 agents; horizon 450



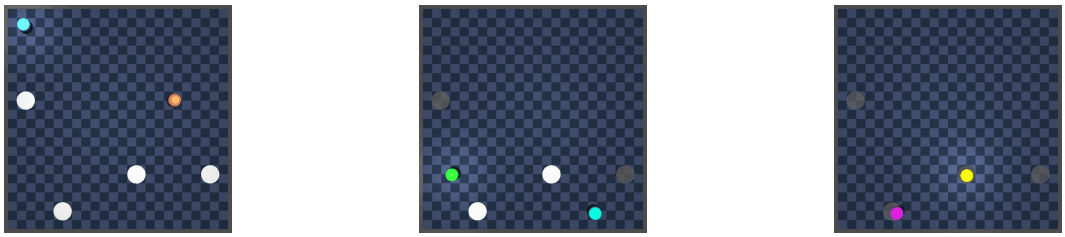
MuJoCo PointMaze3v1Arena
Completion: The designated three-agent or one-agent team wins by reaching its goal first.
Competition; 4 agents; horizon 500



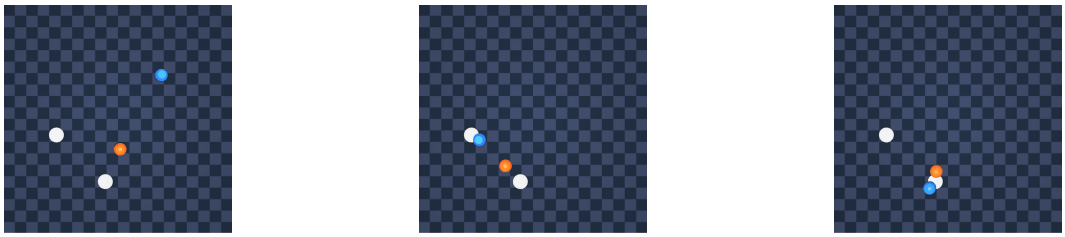
MuJoCo PointMaze3v1Medium
Completion: The same teamwise goal-arrival rule applies in the medium maze layout.
Competition; 4 agents; horizon 500



MuJoCo PointMultiGoalArena
Completion: Both agents finish their two ordered navigation goals.
Multi-Goal; 2 agents; horizon 500



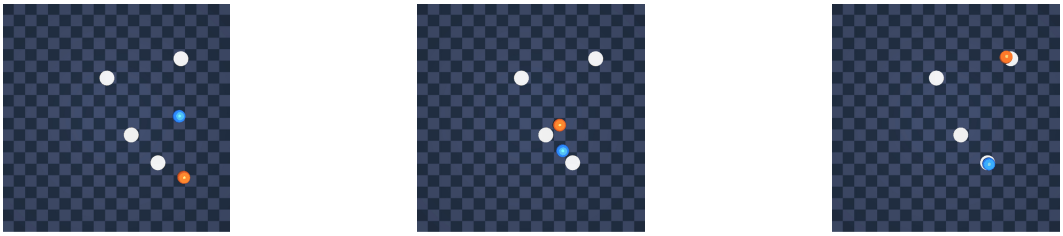
MuJoCo PointMultiGoalOrder
Completion: Each agent enters the first shared-goal region before the second; both must finish.
Multi-Goal; 2 agents; horizon 450



MuJoCo PointMultiGoalParallel

Completion: Both agents complete their own goal sequences, which may advance independently.

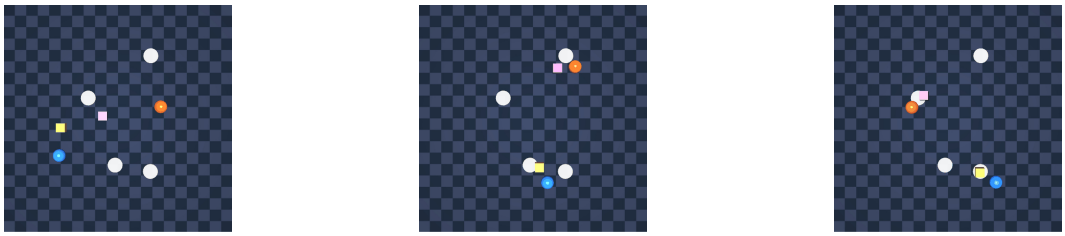
Multi-Goal; 2 agents; horizon 450



MuJoCo PointPushMultiGoal

Completion: Both agents finish all ordered object-pushing goals.

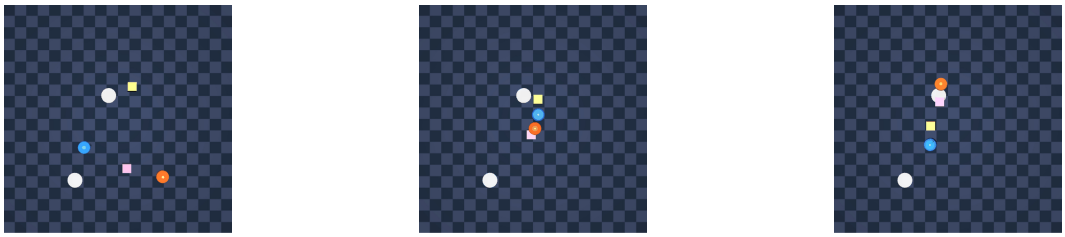
Multi-Goal; 2 agents; horizon 450



MuJoCo PointPushRace

Completion: The first agent to push its own object to its target wins.

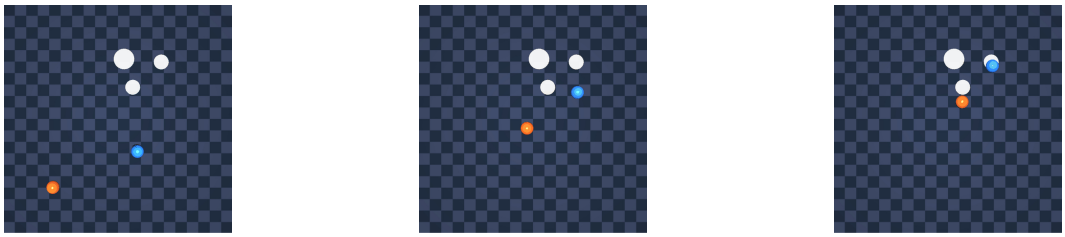
Competition; 2 agents; horizon 300



MuJoCo PointReachAvoidRace

Completion: The first valid goal arrival wins; entering the hazard invalidates success.

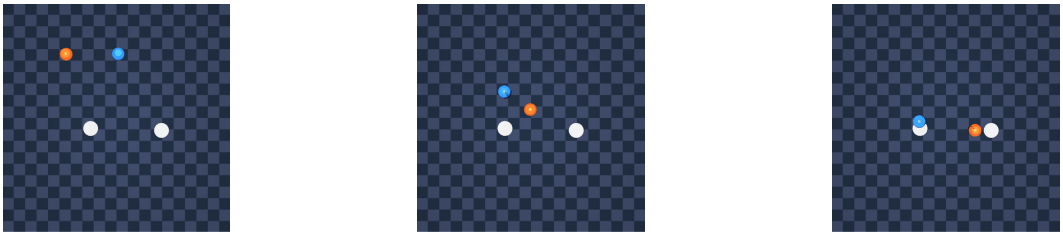
Competition; 2 agents; horizon 300



MuJoCo PointReachRace

Completion: The first valid goal arrival wins, with a draw possible on simultaneous arrival.

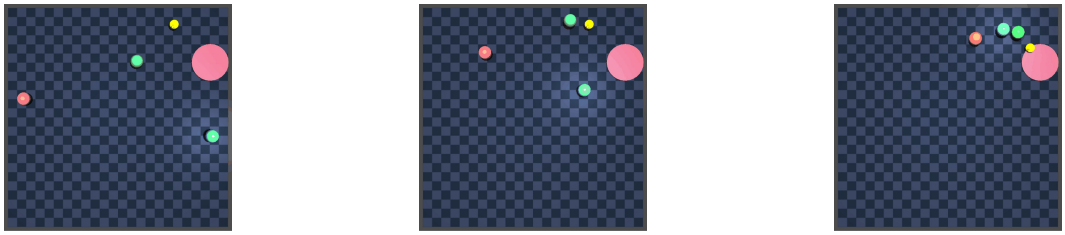
Competition; 2 agents; horizon 300



MuJoCo PointSoccerArena

Completion: Only a goal counts as task success; last-touch credit changes individual reward.

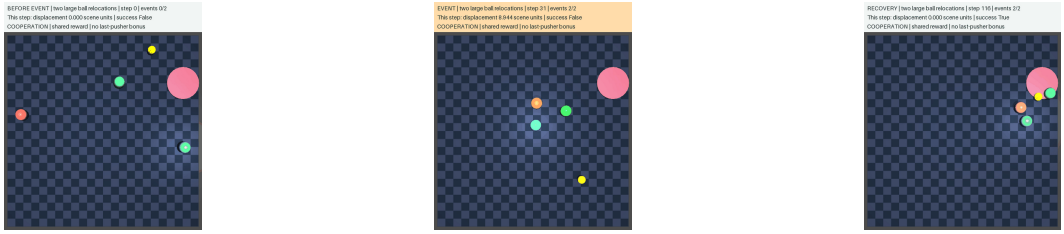
Hybrid; 3 agents; horizon 500



MuJoCo PointSoccerIntervention

Completion: The ball reaches goal; two-intervention recovery additionally requires both events to occur.

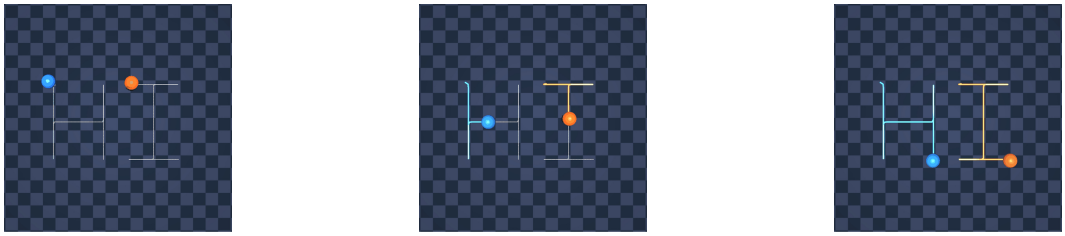
Intervention; 3 agents; horizon 500



MuJoCo PointTrajectoryHI

Completion: Both points traverse all paired H-I waypoints within 0.16 scene units; one complete pattern is counted.

Trajectory-Tracking; 2 agents; horizon 400

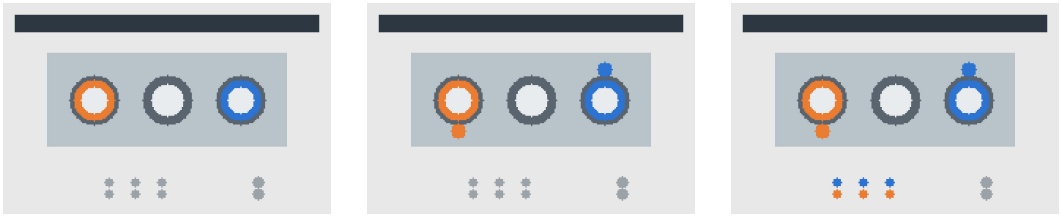


DISCRETE

Discrete CompetitiveButtonRace

Completion: The first agent to achieve three valid progress events wins; simultaneous completion is a draw.

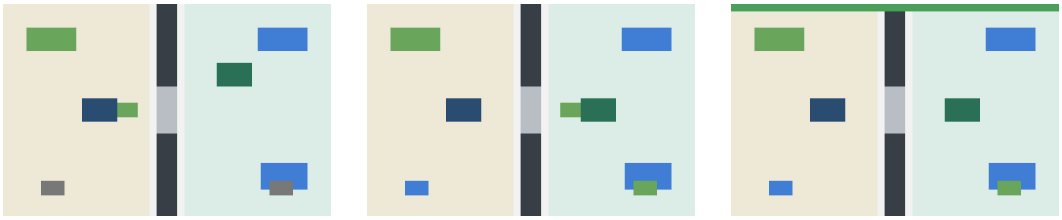
Competition; 2 agents; horizon 24



Discrete KeyedHandoverProtocol

Completion: Code, communication, and synchronized assembly conditions are all satisfied.

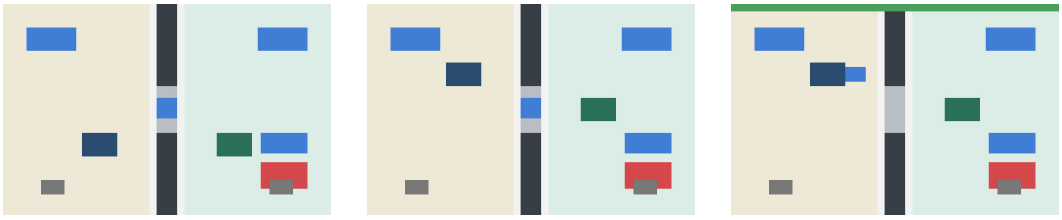
Transition; 2 agents; horizon 18



Discrete KeyedHandoverProtocolCompetitive

Completion: The winning agent submits the shared workpiece; unsubmitted episodes can end in a draw.

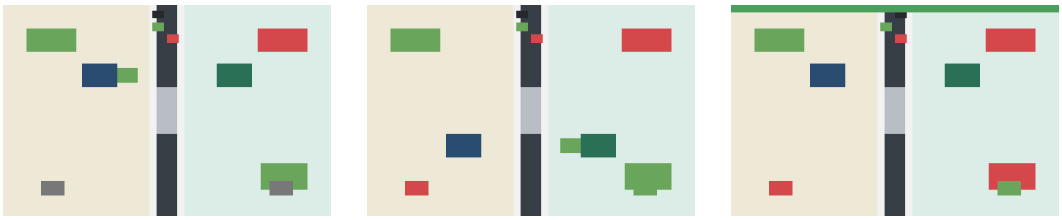
Competition; 2 agents; horizon 18



Discrete KeyedHandoverProtocolMultiGoal

Completion: All assembly goals are completed in sequence; the first alone is intermediate.

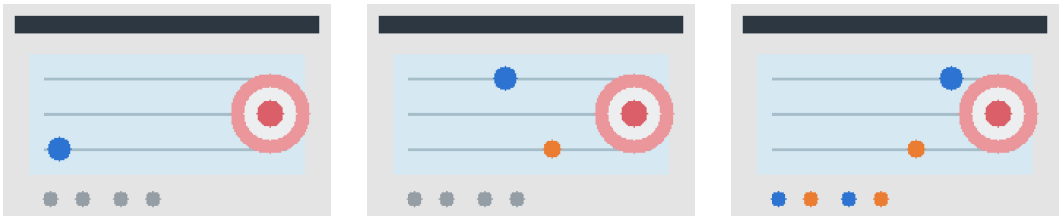
Multi-Goal; 2 agents; horizon 24



Discrete MixedCurling

Completion: Both rounds of valid joint launch-and-sweep operations are completed.

Hybrid; 2 agents; horizon 32



Discrete Stack

Completion: After valid preparation, synchronized PICK, STACK, and RELEASE complete the task.

Cooperation; 2 agents; horizon 28

