# AGENTIC ORCHESTRATION POLICY: ASYNCHRONOUS REASONING AND CONTROL FOR EMBODIED CODING AGENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Coding agents can construct and revise robot behavior during execution, but their deliberation takes time. A single reasoning process can delay work that different robot components could perform independently. We introduce Agentic Orchestration Policy (AOP), an asynchronous framework of persistent coding agents that reason and act for configurable robot control groups. A hyper-agent coordinates their work, deciding what to delegate, which actions must wait for others, and when to control coupled groups directly. This lets local work progress concurrently while coordination adapts to physical dependencies. Across 72 real-world rollouts in 25 task/platform settings, AOP reduces task-balanced time cost relative to Direct Policy in all three categories: Fully Parallel, Parallel-Sequential, and Collaborative, with a 28.2% reduction overall. On Collaborative tasks, it also has lowers mean time cost than a fixed robot arm assignment, which we call fixed delegation. Case studies show how members prepare actions in parallel and how fixed delegation can delay shared actions through repeated planning and waiting. We will release the framework code and a rollout dataset containing observations, generated programs, actions and feedback to support research on how coding agents reason and act in the physical world.

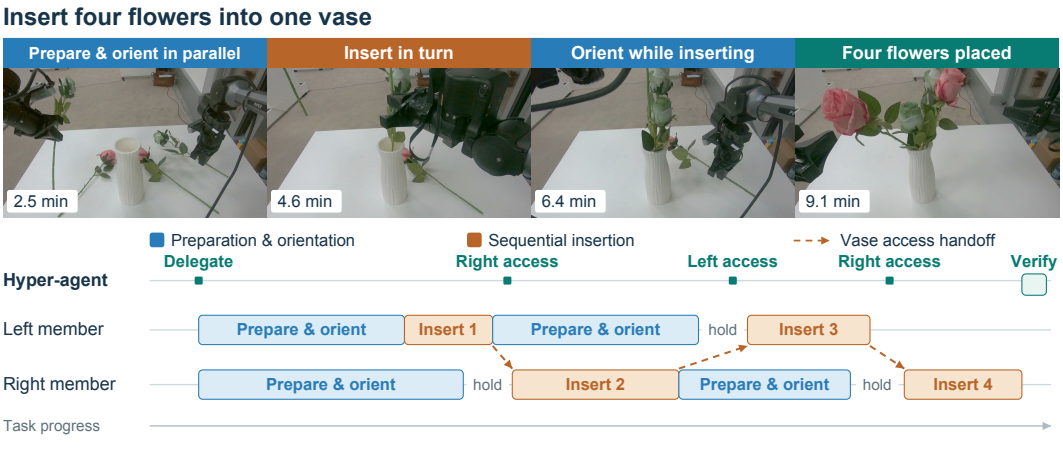



Figure 1: **Parallel preparation, coordinated vase access.** Blue blocks show independent preparation and orientation; orange blocks show insertion at the shared vase. The hyper-agent delegates work and passes vase access between members, so preparation can continue while another flower is inserted. Arrows show the chosen insertion order.[1]

[1] **We strongly recommend viewing the supplementary webpage for side-by-side video comparisons of robot execution and agent coordination.**



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

Recent coding-agent policies can inspect a scene, construct numerical models, and write and revise robot-control programs during execution (Jia et al., 2026; Fu et al., 2026). They extend the executable-policy interface of Code as Policies (Liang et al., 2023) into sustained interaction with physical feedback. While learning from demonstrations supplies increasingly capable visuomotor control (Chi et al., 2023; Kim et al., 2024; Black et al., 2024), coding agents assemble task-specific procedures within an episode. This ability to adapt behavior through computation opens a practical question: *how can coding agents turn their capabilities into efficient physical work?*

Reasoning takes time. A direct coding agent can issue concurrent robot commands, but decisions about different robot components still pass through one reasoning process. When clearing a table, diagnosing one grasp can delay planning an independent pickup. Separate local agents can advance these decisions concurrently, each observing, reasoning and acting at its own pace. Their narrower responsibilities can also accommodate lighter reasoning while a capable agent considers the whole task.

We introduce Agentic Orchestration Policy (AOP), a framework for *asynchronous multi-agent robot policies*. Persistent members serve configurable control groups, such as an arm with its gripper or a mobile robot's torso and chassis. Each member can plan, execute and recover while others continue their work. This distributes both reasoning and action across the embodiment, creating opportunities for parallel progress beyond concurrent motor commands.

When robot components share an object or workspace, their actions must be coordinated. A straightforward design assigns each control group permanently to a member and lets a lead organize their work. This can suit independent subtasks, but folding cloth, lifting a loaded container and opening a bottle require compatible decisions about shared support, contact and motion. Every joint correction must pass through the separate members. Repeated preparation, reporting and authorization can delay progress; one member's recovery can also change the physical conditions assumed by another. More agents alone do not resolve these dependencies.

To adapt the division of work to these dependencies, we introduce a *hyper-agent*, an agent of agents that orchestrates their work. It decides what to delegate, which actions must wait for others, and when to control coupled groups directly. Coordination becomes part of the policy during execution. In four-flower insertion (Figure 1), the coordinating agent alternates vase access while members prepare the next stems independently. For a shared fold, it can instead bring both arms under direct control while retaining member analysis. The aim is to benefit from concurrent local work while preserving the choice of how to execute a coupled action.

We evaluate the efficiency of the complete framework and the value of agentic coordination in two real-world comparisons. Across Fully Parallel, Parallel-Sequential, and Collaborative tasks, we compare AOP with Direct. On Collaborative tasks, we additionally compare AOP with Fixed Delegation to assess agentic coordination against permanent member control, retaining Direct as a reference. Camera sequences and event traces show how local work overlaps and how agents handle shared actions. Generated programs further expose how observations become numerical plans, motivating a dataset for studying embodied coding agents.

Our contributions are threefold:

1. **An asynchronous agentic policy framework.** Coding agents independently observe, reason and act for configurable robot control groups, allowing local work to progress concurrently.
2. **Agentic coordination.** A hyper-agent chooses the members' objectives, dependencies and synchronization, and when to execute a shared action directly.
3. **A real-world rollout dataset.** We will release our dataset containing observations, generated programs, actions and feedback for research on how coding agents model, plan and act in the physical world.

# 2 RELATED WORK

**Executable policies and physical reasoning.** Diffusion Policy, OpenVLA and $\pi_0$ establish effective visuomotor learning and increasingly broad language-conditioned control (Chi et al., 2023;

Kim et al., 2024; Black et al., 2024). Language-model policies make task reasoning explicit: SayCan grounds plans in skill affordances, Inner Monologue incorporates execution feedback, and ReAct interleaves reasoning with action (Ichter et al., 2023; Huang et al., 2023b; Yao et al., 2023). Code as Policies, ProgPrompt and RoboScript express robot behavior through programs (Liang et al., 2023; Singh et al., 2023; Chen et al., 2024a); VoxPoser and ReKep turn spatial objectives into computable value maps and constraints (Huang et al., 2023a; 2024). Agent as Policy and CaP-X extend this direction to interactive coding agents (Jia et al., 2026; Fu et al., 2026). Harness VLA (Zhang et al., 2026) composes fixed analytic primitives with a frozen VLA for contact-rich execution. AOP instead lets coding agents write and revise task-specific models and motion programs, and studies how to coordinate this work across persistent agents.

**Agent workflows and reasoning allocation.** AutoGen, MetaGPT and AgentVerse organize language agents through communication and specialized roles; GPTSwarm represents their interactions as computational graphs (Wu et al., 2023; Hong et al., 2024; Chen et al., 2024b; Zhuge et al., 2024). In robotics, MALMM uses a supervisor to select planning, coding and execution from observations and conversation history (Singh et al., 2025). AOP gives persistent members their own physical observation–reasoning–action loops. Their local analysis can continue when the lead takes control of their groups. FrugalGPT and RouteLLM allocate queries across models with different cost–quality tradeoffs (Chen et al., 2023; Ong et al., 2025); our roles can use different reasoning configurations throughout an episode. We evaluate this allocation jointly with policy organization. Studies of multi-agent failures further motivate examining coordination costs (Cemri et al., 2025).

**Communication and asynchronous embodied agents.** RoCo uses robot dialogue and feedback to refine plans and waypoints; SMART-LLM decomposes tasks and allocates work across robots (Mandi et al., 2024; Kannan et al., 2024). CoELA combines communication with perception, memory and execution, and COMBO models cooperation through compositional world models (Zhang et al., 2024a;b). EMOS combines embodiment-aware group discussion with asynchronous decentralized execution (Chen et al., 2025). AOP focuses on persistent coding agents sharing one embodiment, where a common object may require decisions spanning several control groups. The coordinating agent can preserve local execution, synchronize the participating work, or return those groups to direct lead control.

**Coordination within a robot.** Bimanual taxonomies distinguish spatial and temporal dependencies, while learned role assignment can separate stabilizing and acting arms (Krebs & Asfour, 2022; Grannen et al., 2023). LABOR selects coordination patterns, DAG-Plan schedules arm work through dependency graphs, and LLM+MAP generates partial-order plans using symbolic planning (Chu et al., 2024; Gao et al., 2024; Chu et al., 2025). BiCICLe assigns an LLM to each arm and conditions a follower trajectory on the leader's prediction; Arms' Debate refines these trajectories with fresh in-context prompts (Palma et al., 2026). Our policy organizes persistent coding sessions during execution: objectives, dependencies and the choice of who commands each group are revised from feedback. Appendix F compares these mechanisms. Our experiments compare policy organizations within a common experimental framework, rather than rank complete prior systems.

## 3 Agentic Orchestration Policy

AOP separates two questions: *how can robot policy agents progress independently, and how should their work be coordinated?* The asynchronous framework provides independent reasoning and execution; a hyper-agent chooses the division of work and its dependencies (Figure 2).

### 3.1 Asynchronous reasoning and execution

Let $\mathcal{C}$ be the configured set of robot control groups. A group bundles actuators that can be assigned together, such as an arm and gripper or a torso and chassis. The framework provides a lead $\mathrm{L}$ and a persistent member $\mathrm{M}_c$ for each group $c$. Members are activated when needed and retain their conversations, observations and generated programs across assignments.

Each active agent runs its own observation–reasoning–action loop. One member can diagnose a grasp while another plans a placement; neither must finish its reasoning turn for the other to con-

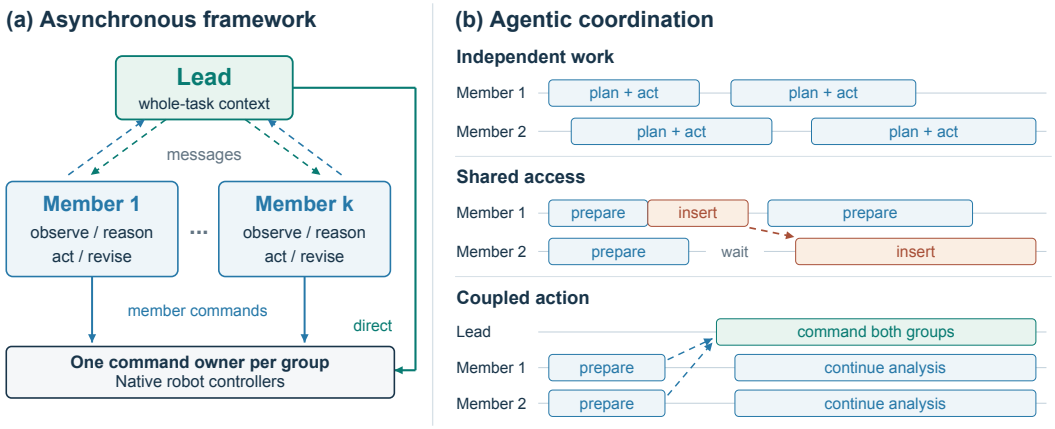



Figure 2: **Independent policy loops with agentic coordination.** (a) Members reason and act for configurable control groups and report to the lead. For a wheeled robot, groups may be the left arm and gripper, right arm and gripper, and torso and chassis. Only the current owner commands each group. (b) The hyper-agent chooses independent work, ordered shared access, or direct execution of coupled actions. The shared-access arrow denotes a dependency coordinated by the lead. These choices adapt during execution; member analysis can continue under direct lead control.

tinue. Low-level controllers track the commanded motions. Thus both deliberation and physical execution can overlap. Members report to the lead, which sends objectives, revisions and relevant evidence. Messages remain available until acknowledged, so progress need not depend on simultaneous reasoning rounds. Narrower local responsibilities can use lighter reasoning while the coordinating agent considers the whole task.

### 3.2 Agentic coordination with a hyper-agent

A *hyper-agent* is the coordination role that reasons about other agents' work. At decision $k$, it uses its interaction history $h_k$, including observations, reports and computed results, and the current command owners $\omega_k$:

$$u_k \sim \pi_H(\cdot \mid h_k, \omega_k), \qquad \omega_k(c) \in \{\mathrm{L}, \mathrm{M}_c\}, \quad c \in \mathcal{C}. \tag{1}$$

Here $u_k$ may assign or revise work, request evidence, specify a dependency, grant shared access, or arrange direct lead execution. The index counts coordination decisions, not common agent rounds. The coding agent produces these decisions from feedback rather than following a supplied task graph.

Delegation gives a member an objective, an allowed scope of action and control of its group. Within that scope, it handles local planning and recovery. As dependencies change, the coordinating agent can keep work independent, sequence member actions at a shared workspace, or return coupled groups to the lead for a joint action. Collaboration can also remain delegated when complementary member roles work well. These choices adapt the organization of the policy, while persistent members preserve their local knowledge.

**Algorithm 1 Asynchronous policy interaction**

Each active agent repeats its own loop; the loops run independently.

**Hyper-agent:**

1. Inspect observations, member reports, pending dependencies and control ownership.
2. Choose assignments, revisions, access grants or direct lead actions from this evidence.
3. Check progress; wait for relevant events, revise coordination, or verify task completion.

**Each member:**

1. Observe and plan for its current objective, incorporating new lead instructions.
2. Execute within its assigned scope and ownership; wait for unmet prerequisites.
3. Report achieved conditions or difficulties with evidence, then continue or await work.

### 3.3 SYNCHRONIZATION AND CONTROL HANDOFF

A synchronization point specifies *which actions must wait and what evidence permits them to continue*. The coordinating agent chooses the participating work and the relevant reports or signals. Agents can wait for any or all specified events for the current assignment, while other agents continue independent work. This forms an event-based barrier where the task needs one; a timeout does not satisfy the dependency. The framework detects the specified events, while the agents interpret readiness and verify the physical condition.

For example, in Figure 1, a member prepares a flower on its own side, then holds it until granted vase access. After the preceding member reports insertion and retreat, the lead checks clearance and grants the next approach. Only vase entry depends on that handoff; preparation can overlap. The same mechanism can collect readiness reports for a shared action. These exchanges coordinate permission to act, while low-level controllers determine physical execution timing.

Command ownership is checked separately. Initially the lead controls each available group; delegation transfers that permission to a member, and taking control back restores it to the lead. This handoff does not stop ongoing motion or establish that an action has finished. The new owner therefore checks physical feedback before acting. Members retain their histories and can continue analysis during lead execution. Algorithm 1 summarizes the resulting interaction.

## 4 REAL-WORLD EVALUATION

### 4.1 TASKS, COMPARISONS AND CONFIGURATION

**Task categories.** All experiments use real robots: a Galaxea R1 Pro 2025 and three AgileX Piper-family installations (R1 and P-A/B/C). The evaluation comprises 72 episodes across 25 task/platform settings, using arms and grippers with the body fixed. Following spatial and temporal distinctions in bimanual manipulation (Krebs & Asfour, 2022; Chu et al., 2024), we distinguish three categories: (i) *Fully Parallel* permits independent local sequences; (ii) *Parallel-Sequential* adds cross-component precedence or shared access; (iii) *Collaborative* couples components through one object's support, motion or deformation. These categories describe task dependencies rather than a prescribed policy schedule.

**Comparisons.** Our two studies correspond to the framework and coordination contributions. All three policies have access to the same observation modalities, numerical tools and robot-control interface:

*(i) Direct Policy (Direct).* One coding agent reasons and controls the whole robot. It can issue concurrent robot commands, but planning and control decisions share one agent context.

*(ii) Fixed Delegation (Fixed).* A lead observes, assigns work, replans and synchronizes permanent control-group members. Only members command their groups. Assignments and timing can change; command ownership cannot return to the lead.

*(iii) Agentic Orchestration (AOP).* The full method in Section 3: the coordinating agent chooses assignments and dependencies, retaining delegated execution or returning groups to the lead when direct coordination is appropriate.

We compare Direct with AOP across all three dependency categories to assess the efficiency of the complete framework. To examine the value of agentic coordination, we additionally compare Fixed with AOP only on Collaborative tasks, where actions must account for shared support, contact or motion. Both methods coordinate member agents, but AOP also lets the lead choose when to execute a coupled action directly. Direct remains a reference for assessing how this flexibility compares with whole-robot control. The comparisons cover 25 and eight common settings, respectively; the Collaborative comparison reuses the corresponding Direct/AOP episodes. Episode details are provided in the appendix.

**Experimental configuration.** We instantiate the hyper-agent role in the lead itself, using the same model and reasoning configuration for orchestration and direct control. Direct uses Astra with the xhigh reasoning setting; Fixed and AOP use an xhigh lead and medium-effort members. Model and reasoning settings remain fixed within an episode. The comparison with Direct includes lighter

member reasoning as part of the framework configuration. The comparison with Fixed holds these settings fixed and changes control permissions and corresponding role instructions. Each rollout starts with fresh agent sessions, without conversations or task-specific programs from earlier rollouts. Operators provide no task guidance or manipulation assistance during execution; they monitor the robot for safety.

### 4.2 Time cost, completion and behavioral evidence

Elapsed time $T_i$ is measured from the start of policy execution to the recorded outcome, or to the end of the episode when no outcome is declared. It includes reasoning, computation, waiting, movement and recovery; task setup before policy execution is excluded. For recorded outcome $o_i$, time cost is

$$L_i = \begin{cases} T_i, & o_i = \text{completed}, \\ \max(T_i, 30\text{ min}), & \text{otherwise}. \end{cases} \tag{2}$$

Thus an unsuccessful early stop incurs at least 30 minutes, while durations beyond 30 minutes remain uncapped. The Fixed bottle failure has no recorded end time and receives the 30-minute floor. Appendix B distinguishes observed durations from time costs.

We average repetitions within each setting, then weight common settings equally. Every evaluated episode contributes; unequal repetition counts do not change a setting's weight. Per-setting counts and individual costs appear in the appendix. Recorded completion at five-minute intervals through 30 minutes uses the same weights and denominators. These are retrospective cutoffs, not uniformly enforced timeouts. Completion follows recorded policy outcomes; physical-quality qualifications are discussed in Appendix B.

Camera images are aligned by recorded capture time and show the earliest episode for each setting and policy. Crops are identical within a setting, and repeated final images are marked. Teaser photographs receive a uniform brightness correction. Comparisons do not assume matched initial object placements. Execution traces link coordination decisions to commands and feedback, without separately attributing time to inference, motion and waiting.

## 5 Results and behavioral analysis

### 5.1 Study 1: efficiency of the asynchronous framework

AOP has lower task-balanced time cost than Direct in each dependency category (Table 1). Its advantage extends from independent work to ordered access and coupled manipulation. A task need not be fully parallel for independent reasoning to help: preparation, perception and local recovery can advance around the actions that require coordination. This result evaluates the complete configuration, including lighter member reasoning.

Table 1: **Study 1: asynchronous framework versus Direct.** Task-balanced mean time cost in minutes; lower is better. Parentheses give settings. Reduction compares AOP with Direct. Every eligible repetition contributes; noncompletion incurs at least 30 minutes.

| Task dependency | Direct | AOP (ours) | Reduction |
|---|---|---|---|
| Fully Parallel (6) | 20.49 | **13.22** | 35.5% |
| Parallel-Sequential (11) | 20.27 | **13.86** | 31.6% |
| Collaborative (8) | 16.25 | **13.75** | 15.4% |
| All settings (25) | 19.04 | **13.67** | 28.2% |

Figure 3 connects these differences to physical progress. Battery collection separates destinations, allowing independent pickups and placements. Hotel preparation overlaps local pickup and reorientation while the lead schedules tray access. Flower insertion makes the dependency particularly clear: a narrow opening requires ordered entry, but preparing the next stem can proceed alongside insertion (Figure 1). In the two-flower trace, the finished member also moves its wrist camera to reveal the other stem's occluded tip. Parallel work remains useful even when the task contains shared resources and changing information needs.

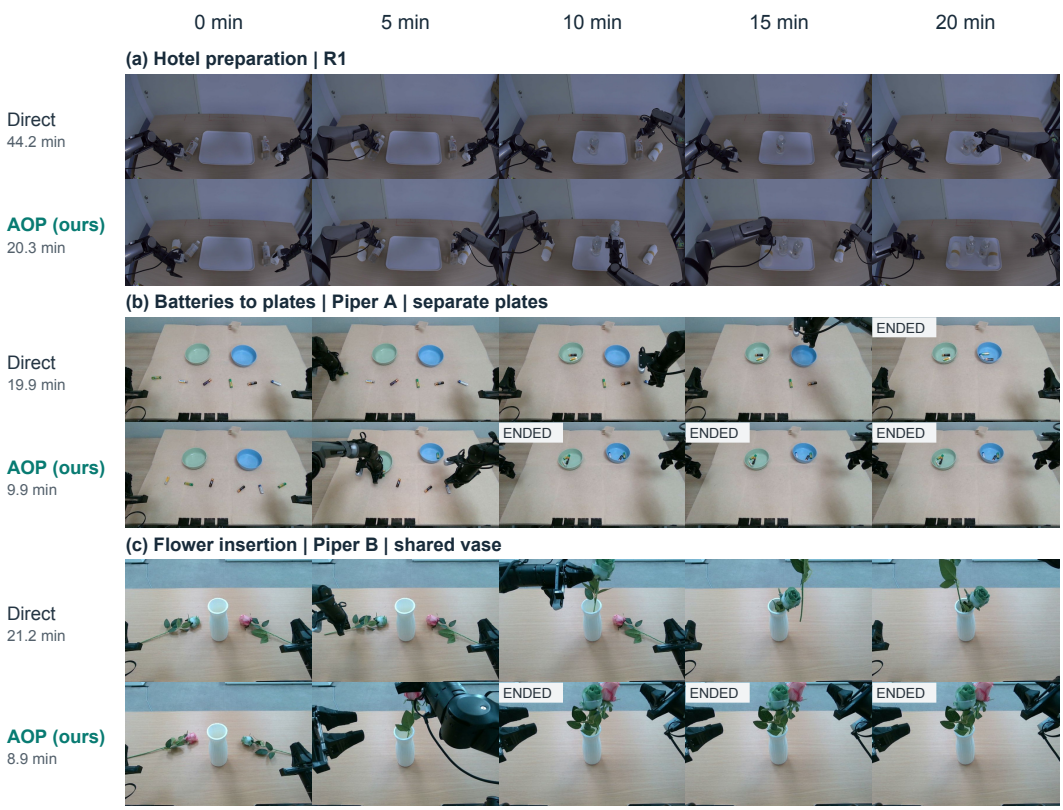



Figure 3: **Study 1: visible progress with Direct and AOP.** Three cases cover Fully Parallel and Parallel-Sequential tasks across R1 and two Piper installations. Columns share elapsed times within each task. Row labels give observed episode durations; ENDED marks repeated final images. The chronological selection and cropping protocol is shared across methods (Section 4).

AOP lowers average time cost across all three task categories. Grid sorting and the R1 basket lift favor Direct, showing that additional coordination can outweigh concurrent work. The aggregate advantage persists when each setting is omitted in turn and when three additional episodes from the same settings are included (Appendix B). These sensitivity analyses assess dependence on corpus composition, not repeated-trial uncertainty.

### 5.2 Study 2: Agentic coordination for collaborative tasks

AOP has lower mean time cost than Fixed in every Collaborative setting and improves on Direct on average (Table 2). Fixed already has lead planning and synchronization. The comparison therefore examines the additional choice available to AOP: whether a shared action should remain distributed across members or be executed directly by the lead. The R1 basket lift still favors Direct; this flexibility is an option, not a guarantee of the fastest execution.

Figure 4 shows two ways to coordinate a shared object. For towel folding, the lead takes over both arms for a shared fold while retaining member analysis. Completion time is close to Direct, whereas Fixed continues through additional preparation and finishing. For bottle opening, AOP keeps stabilization and twisting delegated. Only AOP records completion; both baselines fail after arm crossing triggers hardware faults. These cases illustrate why agentic coordination includes both organizing member actions and deciding when to act directly.

The Fixed traces expose the work involved in distributed coordination. In fruit-basket lifting, about 156 seconds elapse between the preparation instruction and the first motion command, including local computation and checks. In carton lifting, one member's wait condition does not recognize the shared start signal, requiring a later corrective command. Both episodes eventually complete;

Table 2: **Study 2: agentic coordination versus fixed delegation.** Mean time cost in minutes; bold marks the lowest setting mean. Fixed and AOP share agent configurations; Direct is a reference. The final row weights settings equally. †: hardware-fault failure scored at 30 minutes. Both bottle baselines fail; only Fixed lacks a recorded end time.

| Task | Direct | Fixed | AOP (ours) |
|---|---|---|---|
| Ordered towel fold | 14.88 | 34.51 | **14.24** |
| Basket lift | **9.94** | 22.70 | 13.01 |
| Shirt fold | 8.02 | 8.77 | **6.40** |
| Bottle opening | 30.00† | 30.00† | **28.76** |
| Fruit-basket lift | 7.45 | 20.59 | **6.90** |
| Milk-carton lift | 21.13 | 21.88 | **12.28** |
| Pot onto book | 24.03 | 34.85 | **15.63** |
| Basket onto sponge | 14.58 | 34.84 | **12.78** |
| All settings (8) | 16.25 | 26.02 | **13.75** |

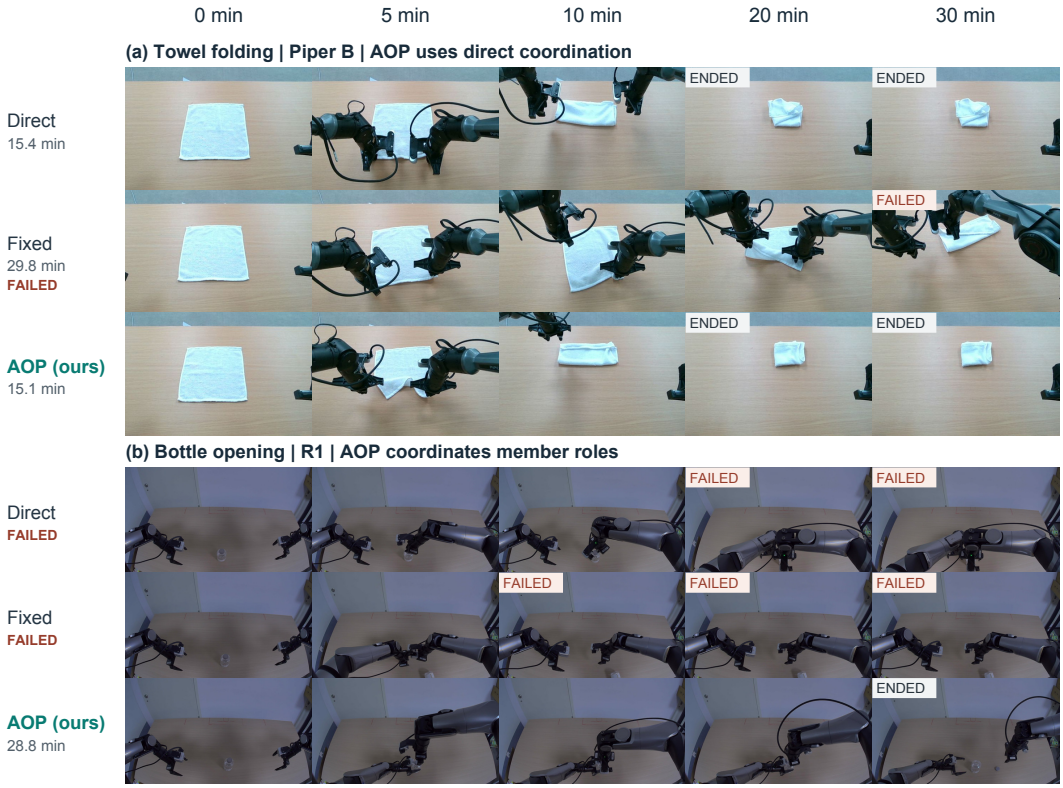



Figure 4: **Study 2: coordination on Collaborative tasks.** Towel folding illustrates direct lead execution within AOP; bottle opening retains complementary member roles. Columns share elapsed times. FAILED denotes no recorded completion. Both bottle baselines trigger hardware faults after arm crossing; later Fixed columns repeat its last available image. Towel-finishing quality is discussed separately.

a revised synchronization routine also reduces the delay between the two motion commands in the basket case. Appendix D analyzes these coordination difficulties and recoveries across all ten Fixed episodes.

**Completion at intermediate budgets.** Table 3 separates lower time cost from earlier goal attainment. Across all settings, AOP records more completions at each horizon after five minutes. On

Table 3: Task-balanced recorded completion (%) by elapsed-time cutoff. Study 1 uses 25 settings; Study 2 uses eight and reuses their Direct/AOP records. Settings and denominators remain fixed across horizons.

| Policy | 5 min | 10 min | 15 min | 20 min | 25 min | 30 min |
|---|---|---|---|---|---|---|
| *Study 1: all 25 settings* | | | | | | |
| Direct | 0.0 | 10.0 | 40.7 | 68.0 | 84.0 | 88.0 |
| AOP (ours) | 0.0 | 22.0 | 74.0 | 86.0 | 96.0 | 100.0 |
| *Study 2: 8 Collaborative settings* | | | | | | |
| Direct | 0.0 | 31.2 | 56.2 | 62.5 | 87.5 | 87.5 |
| Fixed | 0.0 | 12.5 | 12.5 | 18.8 | 43.8 | 43.8 |
| AOP (ours) | 0.0 | 25.0 | 68.8 | 87.5 | 87.5 | 100.0 |

Collaborative tasks, Direct leads at ten minutes, while AOP leads at fifteen and twenty. The average advantage does not imply dominance at every budget.

## 6 A DATASET OF REAL-WORLD CODING-AGENT ROLLOUTS

The rollout dataset links observations, generated programs and numerical results to agent communication, actions, feedback and outcomes. These records expose both how agents construct physical plans and how they distribute the work of carrying them out.

The programs show a recurring process: build a local analytical model, derive an action, and check the object's response. A fitted bottle axis determines a corrective wrist rotation. During carton lifting, a member continues analysis after the lead takes control; tracking material features corrects an apparent-slip diagnosis based on fixed image locations. These examples connect mathematical reasoning to physical decisions and show how member analysis can remain useful during centralized execution. Appendix E traces the computations and their limitations. The dataset supports studying these processes; the present comparisons do not isolate mathematics as the cause of generalization. Release contents are described in Appendix G.

## 7 DISCUSSION AND LIMITATIONS

The results support making the organization of reasoning and control part of an embodied policy. Their scope is a retrospective corpus with limited replication, recurring manipulation families and unmatched initial conditions. Outcome labels and completeness of attempt coverage have not been independently verified. The comparisons describe these settings, without a held-out generalization test or a guarantee of matching Direct on every task. All modes share calibration and kinematic tools; experiments use arms and grippers with the body fixed. Lighter member reasoning and asynchrony are evaluated jointly. Our efficiency results concern elapsed task time while the trade-off with total inference cost remains to be quantified.

Collision avoidance remains a limitation of the current system. The ownership check controls which agent can issue commands, but does not validate geometric clearance. Collisions can therefore occur between robot components or with the environment. Addressing this limitation requires collision-aware motion planning and execution checks in the underlying robot-control layer, which are beyond the scope of our study of asynchronous reasoning and agentic coordination.

## 8 CONCLUSION

AOP combines independent coding-agent policy loops with a hyper-agent that coordinates their work. Across the evaluated tasks, this organization improves time cost over Direct, while flexible coordination improves on permanent member control for Collaborative tasks. The dataset exposes the computational and physical traces behind these decisions for further study.

### AI USE STATEMENT

The core research ideas were developed by the human authors. Generative AI tools, including OpenAI Codex, assisted literature retrieval, drafting and polishing manuscript sections, implementation, and data analysis. AI also participates in research execution as the object of study: coding agents generate and execute robot-control programs during the real-world experiments. Their model and reasoning settings are specified in Section 4. AI was not used to generate synthetic experimental datasets or prove mathematical claims. The authors take responsibility for the final manuscript and for verifying its claims, analyses and supporting evidence.

### ETHICS STATEMENT

Human operators continuously supervise the robot during rollouts and remain ready to stop unsafe operation to prevent unwanted damage, including damage associated with hardware faults. This safety supervision provides no task guidance or manipulation assistance. Identifiable content will be blurred in videos and case-study images before public release.

### REPRODUCIBILITY STATEMENT

Sections 3–4 define the policy variants and evaluation metrics. Appendix B reports per-episode measurements; displayed images are linked to their original recordings and capture times. Appendix G describes the supplied information and role instructions. We will release the framework, rollout dataset and analysis needed to reproduce the quantitative results.

## REFERENCES

Kevin Black, Noah Brown, Danny Driess, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why Do Multi-Agent LLM Systems Fail? In *Advances in Neural Information Processing Systems 38*, pp. 135676–135729, 2025. doi: 10.52202/085713-4082. URL https://arxiv.org/abs/2503.13657.

Junting Chen, Yao Mu, Qiaojun Yu, Tianming Wei, Silang Wu, Zhecheng Yuan, Zhixuan Liang, Chao Yang, Kaipeng Zhang, Wenqi Shao, Yu Qiao, Huazhe Xu, Mingyu Ding, and Ping Luo. RoboScript: Code Generation for Free-Form Manipulation Tasks across Real and Simulation. *arXiv preprint arXiv:2402.14623*, 2024a. URL https://arxiv.org/abs/2402.14623.

Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, Yao Mu, Mengkang Hu, Wenqi Shao, Yikai Wang, Guohao Li, and Lin Shao. EMOS: Embodiment-aware heterogeneous multi-robot operating system with LLM agents. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ey8KcabBpB.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv preprint arXiv:2305.05176*, 2023. URL https://arxiv.org/abs/2305.05176.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=EHg5GDnyq1.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, Wenhao Lu, and Stefan Wermter. Large Language Models for Orchestrating Bimanual Robots. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pp. 328–334, 2024. doi: 10.1109/humanoids58906.2024.10769891. URL https://arxiv.org/abs/2404.02018.

Kun Chu, Xufeng Zhao, Cornelius Weber, and Stefan Wermter. LLM+MAP: Bimanual Robot Task Planning using Large Language Models and Planning Domain Definition Language. *arXiv preprint arXiv:2503.17309*, 2025. URL https://arxiv.org/abs/2503.17309.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Fei-Fei Li, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi "Jim" Fan. CaP-X: A framework for benchmarking and improving coding agents for robot manipulation. *arXiv preprint arXiv:2603.22435*, 2026. URL https://arxiv.org/abs/2603.22435v2. Version 2.

Zeyu Gao, Yao Mu, Jinye Qu, Mengkang Hu, Lingyue Guo, Ping Luo, and Yanfeng Lu. DAG-Plan: Generating Directed Acyclic Dependency Graphs for Dual-Arm Cooperative Planning. *arXiv preprint arXiv:2406.09953*, 2024. URL https://arxiv.org/abs/2406.09953v1.

Jennifer Grannen, Yilin Wu, Brandon Vu, and Dorsa Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 563–576, 2023. URL https://proceedings.mlr.press/v229/grannen23a.html.

Sirui Hong, Mingchen Zhuge, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562, 2023a. URL https://proceedings.mlr.press/v229/huang23b.html.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782, 2023b. URL https://proceedings.mlr.press/v205/huang23c.html.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation. *arXiv preprint arXiv:2409.01652*, 2024. URL https://arxiv.org/abs/2409.01652.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Mengzhao Jia, Yang Lin, Xixin Zhang, Zhihan Zhang, Xiaobai Liu, and Meng Jiang. Agent as policy for robotic manipulation. *arXiv preprint arXiv:2609.12541*, 2026. URL https://arxiv.org/abs/2609.12541v2. Version 2.

Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. SMART-LLM: Smart Multi-Agent Robot Task Planning using Large Language Models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12140–12147, 2024. doi: 10.1109/iros58592.2024.10802322. URL https://arxiv.org/abs/2309.10062.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Franziska Krebs and Tamim Asfour. A Bimanual Manipulation Taxonomy. *IEEE Robotics and Automation Letters*, pp. 11031–11038, 2022. doi: 10.1109/lra.2022.3196158. URL https://doi.org/10.1109/lra.2022.3196158.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023. doi: 10.1109/ICRA48891.2023.10160591. URL https://arxiv.org/abs/2209.07753.

Zhao Mandi, Shreeya Jain, and Shuran Song. RoCo: Dialectic Multi-Robot Collaboration with Large Language Models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299, 2024. doi: 10.1109/icra57147.2024.10610855. URL https://arxiv.org/abs/2307.04738.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route llms from preference data. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Alessio Palma, Indro Spinelli, Vignesh Prasad, Luca Scofano, Yufeng Jin, Georgia Chalvatzaki, and Fabio Galasso. Bimanual Robot Manipulation via Multi-Agent In-Context Learning. *arXiv preprint arXiv:2604.20348*, 2026. URL https://arxiv.org/abs/2604.20348v1.

Harsh Singh, Rocktim Jyoti Das, Mingfei Han, Preslav Nakov, and Ivan Laptev. MALMM: Multi-Agent Large Language Models for Zero-Shot Robotic Manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20386–20393, 2025. doi: 10.1109/iros60139.2025.11247340. URL https://doi.org/10.1109/iros60139.2025.11247340.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530, 2023. doi: 10.1109/icra48891.2023.10161317. URL https://arxiv.org/abs/2209.11302.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*, 2023. URL https://arxiv.org/abs/2308.08155.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building Cooperative Embodied Agents Modularly with Large Language Models. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=EnXJfQqy0K.

Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. COMBO: Compositional World Models for Embodied Multi-Agent Cooperation. *arXiv preprint arXiv:2404.10775*, 2024b. URL https://arxiv.org/abs/2404.10775.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Yi Nie, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Jiaji Rao, Zhengru Fang, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026. URL https://arxiv.org/abs/2607.08448.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

# A TASK DEFINITIONS AND CORPUS

Table 4 lists all 25 experimental task/platform settings. R1 denotes Galaxea R1 Pro 2025; P-A/B/C denote three AgileX Piper-family installations. A setting is a task on one installation, not an independent manipulation skill family. Flowers, batteries and basket lifting recur across installations; P-C also includes distinct two-flower and four-flower configurations. The installations therefore demonstrate applicability without constituting a controlled morphology-transfer study.

We use the same dependency categories as Section 4: *Fully Parallel*, *Parallel-Sequential*, and *Collaborative*. Fully Parallel permits sequential actions within each control group's workload. Hotel service is categorized by the intended independence of its two side workloads: local bottle-before-towel ordering permits both sides to progress concurrently. The actual coordination depends on conditions encountered during execution. Shared-workspace avoidance and recovery can introduce further serialization without changing the intended task category.

Table 4: Task instructions used in the evaluation. Basket lifting includes two instruction variants represented once per mode.

| ID | Instruction |
|---|---|
| **Fully Parallel** | |
| T01 | Put the batteries into their nearest plates. |
| T02 | Put the batteries into their nearest box. |
| T03 | Clean up the dirty water with the blue towel. Put the clean white towel rolls into the tray. |
| T04 | Put the two bottles of water into the tray, upright in the center, then put the rolled towels onto the tray, one on each side. |
| T05 | Put the towel rolls into the tray, side by side. |
| T06 | Put the cubes into the plate, batteries into the cup. |
| **Parallel-Sequential** | |
| T07 | Put the pens into the pen holder. |
| T08 | Insert the plastic flowers into the vase. |
| T09 | Put the items in respective grids. |
| T10 | Insert the plastic flowers into the vase. |
| T11 | Put the pens into the pen holder. |
| T12 | Put the clothes into the basket. |
| T13 | Put the sponge cubes into the tank. |
| T14 | Stack the paper cups to the one marked with X. |
| T15 | Put the toy blocks on the table into the basket. |
| T16 | Stack the towel rolls into a pyramid on the tray, three at the bottom, two in the middle, and one on top. |
| T17 | Insert the plastic flowers into the vase. |
| **Collaborative** | |
| T18 | Fold the towel into a quarter square, from bottom to top, then from left to right. |
| T19 | Lift the basket in the air, and hold still. / Lift the basket in the air, with a clear margin above the table, and hold still. |
| T20 | Fold the T-shirt into a square, from bottom to top. |
| T21 | Twist open the bottle. |
| T22 | Lift the fruit basket in the air, with a clear margin above the table, and hold still. |
| T23 | Hold the milk brick in the air with both hands, showing a clear margin above the table. |
| T24 | Put the pot on the book. |
| T25 | Put the fruit basket onto the black sponge. |

The evaluation comprises 72 episodes: 31 Direct, 31 AOP and ten Fixed. Of these, 65 have recorded completion and seven do not complete: six have recorded termination times, and one hardware-fault failure has no recorded end time. Model and reasoning settings are given in Section 4.

These episodes form the primary evaluation cohort. Additional episodes from the same settings are considered in the sensitivity analysis below; other exploratory and incomplete attempts are outside the primary comparison.

# B Complete recorded results and sensitivity

## B.1 Replication and coverage

Table 5 reports repetition counts. Direct and AOP have equal counts within each of the 25 settings, while replication varies across settings. Task IDs T01–T25 follow the category order in Table 4. All eight Collaborative settings have all three modes.

Table 5: Episode counts by task category and comparison scope. Fully Parallel and Parallel-Sequential compare Direct with AOP; the Fixed comparison is conducted on Collaborative tasks. The main comparison contains 31 episodes per mode; the Collaborative subset contains ten per mode, including ten additional Fixed episodes.

| **Fully Parallel** | | | |
|---|---|---|---|
| ID | Setting (platform) | Direct | AOP |
| T01 | Batteries to plates (P-A) | 2 | 2 |
| T02 | Batteries to boxes (P-C) | 1 | 1 |
| T03 | Wipe and arrange towels (R1) | 1 | 1 |
| T04 | Hotel tray preparation (R1) | 2 | 2 |
| T05 | Towel rolls to tray (R1) | 3 | 3 |
| T06 | Cubes and batteries sorting (P-C) | 1 | 1 |
| Category total | | 10 | 10 |

| **Parallel-Sequential** | | | |
|---|---|---|---|
| ID | Setting (platform) | Direct | AOP |
| T07 | Pens to holder (P-A) | 1 | 1 |
| T08 | Flowers to vase (P-B) | 1 | 1 |
| T09 | Grid sorting (P-B) | 1 | 1 |
| T10 | Flowers to vase (P-C) | 1 | 1 |
| T11 | Pens to holder (P-C) | 1 | 1 |
| T12 | Clothes to basket (R1) | 1 | 1 |
| T13 | Sponge cubes to tank (R1) | 1 | 1 |
| T14 | Cups to marked cup (P-B) | 1 | 1 |
| T15 | Toy blocks to basket (P-B) | 1 | 1 |
| T16 | Towel-roll pyramid (P-C) | 1 | 1 |
| T17 | Four flowers to vase (P-C) | 1 | 1 |
| Category total | | 11 | 11 |

| **Collaborative** | | | | |
|---|---|---|---|---|
| ID | Setting (platform) | Direct | Fixed | AOP |
| T18 | Ordered towel fold (P-B) | 2 | 2 | 2 |
| T19 | Basket lift (R1) | 2 | 2 | 2 |
| T20 | Shirt fold (P-C) | 1 | 1 | 1 |
| T21 | Bottle opening (R1) | 1 | 1 | 1 |
| T22 | Fruit-basket lift (P-A) | 1 | 1 | 1 |
| T23 | Milk-carton lift (P-B) | 1 | 1 | 1 |
| T24 | Pot onto book (P-A) | 1 | 1 | 1 |
| T25 | Basket onto sponge (P-B) | 1 | 1 | 1 |
| Category total | | 10 | 10 | 10 |

### B.2 TIME COST AND OBSERVED DURATION

Equation 2 assigns at least 30 minutes to a noncompleted episode while retaining larger observed durations. The penalty is not a claim of eventual completion or a measured 30-minute execution. Table 6 gives every time cost; Table 7 separately reports observed duration and outcome. Every episode in the primary cohort contributes. The sensitivity analysis below examines the effect of including additional episodes from the same settings.

Table 6: All individual time costs (minutes), grouped by the same comparison scope as Table 5. Superscript F marks noncompletion and includes the failure floor; unmarked values have recorded completion. Fixed is evaluated in the Collaborative panel. Raw times and outcome labels are preserved in Table 7.

**Fully Parallel**

| ID | Task (platform) | Direct | AOP |
|---|---|---|---|
| T01 | Batteries to plates (P-A) | 19.89, 17.73 | 9.93, 14.69 |
| T02 | Batteries to boxes (P-C) | 21.26 | 12.50 |
| T03 | Wipe and arrange towels (R1) | 15.56 | 14.46 |
| T04 | Hotel tray preparation (R1) | 44.17, 32.17 | 20.29, 18.06 |
| T05 | Towel rolls to tray (R1) | 13.93, 18.43, 12.05 | 11.16, 10.40, 13.04 |
| T06 | Cubes and batteries sorting (P-C) | 14.34 | 9.35 |

**Parallel-Sequential**

| ID | Task (platform) | Direct | AOP |
|---|---|---|---|
| T07 | Pens to holder (P-A) | 18.06 | 13.03 |
| T08 | Flowers to vase (P-B) | 21.25 | 8.87 |
| T09 | Grid sorting (P-B) | 11.37 | 15.21 |
| T10 | Flowers to vase (P-C) | 55.06 | 22.53 |
| T11 | Pens to holder (P-C) | 14.10 | 11.38 |
| T12 | Clothes to basket (R1) | 26.53 | 24.49 |
| T13 | Sponge cubes to tank (R1) | 13.35 | 11.25 |
| T14 | Cups to marked cup (P-B) | 18.55 | 10.71 |
| T15 | Toy blocks to basket (P-B) | 18.70 | 13.62 |
| T16 | Towel-roll pyramid (P-C) | 15.62 | 12.17 |
| T17 | Four flowers to vase (P-C) | 10.34 | 9.16 |

**Collaborative**

| ID | Task (platform) | Direct | Fixed | AOP |
|---|---|---|---|---|
| T18 | Ordered towel fold (P-B) | 15.44, 14.33 | $30.00^{F}$, $39.01^{F}$ | 15.05, 13.42 |
| T19 | Basket lift (R1) | 11.92, 7.95 | 15.41, $30.00^{F}$ | 13.34, 12.68 |
| T20 | Shirt fold (P-C) | 8.02 | 8.77 | 6.40 |
| T21 | Bottle opening (R1) | $30.00^{F}$ | $30.00^{F}$ | 28.76 |
| T22 | Fruit-basket lift (P-A) | 7.45 | 20.59 | 6.90 |
| T23 | Milk-carton lift (P-B) | 21.13 | 21.88 | 12.28 |
| T24 | Pot onto book (P-A) | 24.03 | $34.85^{F}$ | 15.63 |
| T25 | Basket onto sponge (P-B) | 14.58 | $34.84^{F}$ | 12.78 |

**Bottle opening: hardware failures and time cost.** Only AOP completes bottle opening. The experimenter confirms that Direct and Fixed fail when their arms cross and collide, triggering hardware faults. The Fixed record describes an overtemperature fault with the arms locked together; the Direct record describes an overload after arm crossing. Both therefore receive the 30-minute failure cost, and neither contributes a completion at any horizon.

Direct has a recorded termination at 15.90 minutes; Fixed has no recorded end time. The last available Fixed camera image is at 8.65 minutes, and later columns repeat this image with a FAILED marker. The image's capture time is not treated as the episode's end time. We also evaluate the

aggregate result without bottle opening to assess its dependence on this setting. Because AOP retains member control here, the case supports effective collaborative orchestration without attributing success specifically to direct takeover.

**Equal task weighting.** We average repetitions within each setting and then average the setting means. All shared settings receive equal weight; the three categories do not. This corrects unequal representation of tasks without suggesting that extra repeats remove uncertainty about a sparsely observed setting. Frames, sensor samples and tool calls are not additional experimental units.

**Recorded completion.** At each horizon, we average the completed-by-horizon indicator within each setting, then average settings equally. Denominators remain fixed across horizons. Failed and stopped episodes do not become completed at the penalty time. A completion after a horizon contributes zero at that horizon but remains in the dataset. Operator interruptions are not reinterpreted as uniformly enforced timeouts. Recorded completion can differ from physical attainment through either premature acceptance or prolonged finishing. We report these recorded outcomes without assigning an independent physical-quality score; the following cases describe relevant qualifications.

**Cup stacking and physical quality.** AOP cup stacking records completion at 10.71 minutes, versus 18.55 minutes for Direct. Both outcome summaries describe paper-cup damage: creases or dents for Direct, and a visibly creased/torn top cup for AOP. We retain the recorded completion labels under the stated protocol and report the damage in the accompanying outcome annotations. These observations establish neither damage-free success nor equal finishing quality.

**Additional shared-load tasks.** All three modes record completion for fruit-basket and milk-carton lifting. Their outcome summaries describe visible clearance and subsequent holds, supported by image/depth analysis and stationary joint feedback. These policy-generated estimates do not survey the hidden underside or establish equal lift heights. Direct's milk-carton outcome also reports upper-side creasing without visible leakage. These qualifications accompany the per-episode results; Section E analyzes the generated measurement code separately from independent outcome scoring.

### B.3 Sensitivity to replication and task coverage

All evaluated settings have equal Direct/AOP repetition counts, totaling 31 episodes per mode. The primary analysis uses all of these episodes; no equal-count subsampling is required. Settings remain equally weighted because their repetition counts differ.

Removing each setting in turn gives a 24.15–29.81% reduction. Excluding bottle opening retains 24 settings and gives 18.58 versus 13.04 minutes (29.81%). AOP has lower mean time cost in 23 of 25 settings. These sensitivity analyses assess dependence on task coverage; they are not confidence intervals or replacements for repeated trials.

**Sensitivity to additional episodes.** Three earlier episodes from the same settings are outside the primary cohort: AOP pen placement E018 (57.93 minutes), AOP mixed-object sorting E065 (16.14 minutes), and Direct P-C pen placement E026 (11.36 minutes). The primary cohort instead includes E067 (13.03 minutes), E075 (9.35 minutes), and E027 (14.10 minutes), respectively. Including all three earlier episodes as additional repetitions gives task-balanced costs of 18.98 versus 14.70 minutes, a 22.54% reduction; AOP completion at 30 minutes becomes 98.0%. This analysis includes both the slower AOP episodes and the faster Direct episode. The Fixed bottle failure remains included despite its missing end time.

Table 7: Per-episode durations and outcomes. D: Direct; A: AOP; F: Fixed. Raw time is observed time to completion declaration or termination; cost applies the failure floor. The dash denotes an unavailable end time. Outcome labels follow recorded completion, with the two bottle hardware failures identified explicitly. No independent physical-quality score is assigned.

| Episode | Task | Mode | Raw min | Cost min | Outcome |
|---|---|---|---|---|---|
| E001 | T01 | A | 9.93 | 9.93 | completed |
| E002 | T01 | A | 14.69 | 14.69 | completed |
| E003 | T01 | D | 19.89 | 19.89 | completed |
| E004 | T01 | D | 17.73 | 17.73 | completed |
| E005 | T02 | A | 12.50 | 12.50 | completed |
| E006 | T02 | D | 21.26 | 21.26 | completed |
| E007 | T03 | A | 14.46 | 14.46 | completed |
| E008 | T03 | D | 15.56 | 15.56 | completed |
| E009 | T04 | A | 20.29 | 20.29 | completed |
| E010 | T04 | A | 18.06 | 18.06 | completed |
| E011 | T04 | D | 44.17 | 44.17 | completed |
| E012 | T04 | D | 32.17 | 32.17 | completed |
| E013 | T05 | A | 11.16 | 11.16 | completed |
| E014 | T05 | A | 10.40 | 10.40 | completed |
| E015 | T05 | A | 13.04 | 13.04 | completed |
| E016 | T05 | D | 13.93 | 13.93 | completed |
| E017 | T05 | D | 18.43 | 18.43 | completed |
| E019 | T08 | A | 8.87 | 8.87 | completed |
| E020 | T08 | D | 21.25 | 21.25 | completed |
| E021 | T09 | A | 15.21 | 15.21 | completed |
| E022 | T09 | D | 11.37 | 11.37 | completed |
| E023 | T10 | A | 22.53 | 22.53 | completed |
| E024 | T10 | D | 55.06 | 55.06 | completed |
| E025 | T11 | A | 11.38 | 11.38 | completed |
| E027 | T11 | D | 14.10 | 14.10 | completed |
| E028 | T12 | A | 24.49 | 24.49 | completed |
| E029 | T12 | D | 26.53 | 26.53 | completed |
| E030 | T13 | A | 11.25 | 11.25 | completed |
| E031 | T13 | D | 13.35 | 13.35 | completed |
| E035 | T18 | A | 15.05 | 15.05 | completed |
| E036 | T18 | A | 13.42 | 13.42 | completed |
| E037 | T18 | D | 15.44 | 15.44 | completed |
| E038 | T18 | D | 14.33 | 14.33 | completed |
| E039 | T18 | F | 29.83 | 30.00 | failed |
| E040 | T18 | F | 39.01 | 39.01 | failed |
| E041 | T19 | A | 13.34 | 13.34 | completed |
| E042 | T19 | A | 12.68 | 12.68 | completed |
| E043 | T19 | D | 11.92 | 11.92 | completed |
| E044 | T19 | D | 7.95 | 7.95 | completed |
| E045 | T19 | F | 15.41 | 15.41 | completed |
| E046 | T19 | F | 29.75 | 30.00 | failed |
| E047 | T14 | D | 18.55 | 18.55 | completed |
| E048 | T15 | A | 13.62 | 13.62 | completed |
| E049 | T15 | D | 18.70 | 18.70 | completed |
| E050 | T16 | A | 12.17 | 12.17 | completed |
| E051 | T16 | D | 15.62 | 15.62 | completed |
| E052 | T20 | A | 6.40 | 6.40 | completed |
| E053 | T20 | D | 8.02 | 8.02 | completed |
| E054 | T20 | F | 8.77 | 8.77 | completed |
| E055 | T21 | A | 28.76 | 28.76 | completed |

| Episode | Task | Mode | Raw min | Cost min | Outcome |
|---|---|---|---|---|---|
| E056 | T21 | D | 15.90 | 30.00 | failed (hardware) |
| E057 | T21 | F | — | 30.00 | failed (hardware) |
| E058 | T14 | A | 10.71 | 10.71 | completed |
| E059 | T22 | A | 6.90 | 6.90 | completed |
| E060 | T22 | D | 7.45 | 7.45 | completed |
| E061 | T22 | F | 20.59 | 20.59 | completed |
| E062 | T23 | A | 12.28 | 12.28 | completed |
| E063 | T23 | D | 21.13 | 21.13 | completed |
| E064 | T23 | F | 21.88 | 21.88 | completed |
| E066 | T06 | D | 14.34 | 14.34 | completed |
| E067 | T07 | A | 13.03 | 13.03 | completed |
| E068 | T07 | D | 18.06 | 18.06 | completed |
| E069 | T24 | A | 15.63 | 15.63 | completed |
| E070 | T24 | D | 24.03 | 24.03 | completed |
| E071 | T24 | F | 34.85 | 34.85 | failed |
| E072 | T25 | A | 12.78 | 12.78 | completed |
| E073 | T25 | D | 14.58 | 14.58 | completed |
| E074 | T25 | F | 34.84 | 34.84 | failed |
| E075 | T06 | A | 9.35 | 9.35 | completed |
| E076 | T05 | D | 12.05 | 12.05 | completed |
| E077 | T17 | A | 9.16 | 9.16 | completed |
| E078 | T17 | D | 10.34 | 10.34 | completed |

## C QUALITATIVE ANALYSIS AND VISUAL COMPARISONS

Figure 1 illustrates four-flower insertion; Figure 3 compares hotel service, batteries and two-flower insertion with Direct. Figure 4 shows towel folding and bottle opening. These cases cover independent destinations and shared physical dependencies. The teaser illustrates how independent preparation continues around sequential use of the shared vase, with access chosen by the hyper-agent. For each displayed task/mode, the earliest available primary episode is used. Repetitions are not ranked by completion time. The collaborative comparison applies the same rule to all three modes. Initial placements differ, so the rows are not paired-reset experiments.

Images are aligned by capture time relative to the start of policy execution: 0/5/10/15/20 minutes for Study 1 and the towel-pyramid comparison, and 0/5/10/20/30 for Study 2 and the remaining supplementary figures. We select the nearest available image within 30 seconds of the target time or recorded episode end. Repeated final images are marked ENDED for completed episodes and FAILED otherwise. Fixed bottle opening has no recorded end time: its three later columns repeat the last available image at 8.65 minutes and remain labeled FAILED. These repeats are not later observations. Each displayed image is linked to its original recording and capture time. Comparison rows use identical crops across modes within a task. Row labels use observed elapsed times rather than penalized costs; both failed bottle-opening attempts are labeled FAILED. Elapsed time is measured independently of video playback speed. Image alignment relies on the recorded camera and policy clocks; their relative offsets were not independently measured. An identifying region in a supplementary image is blurred without altering the task or robot.

**Illustrating parallel preparation and ordered access.** Figure 1 illustrates AOP four-flower insertion (E077, T17). Its photographs are nearest to 2.5, 4.6, 6.4 and 9.1 minutes, with one fixed crop and no photometric alteration. The schematic shows how each member prepares and orients a flower while the other approaches the shared vase. Insertion proceeds left, right, left, then right, following the episode's observed order. After each insertion and retreat, the hyper-agent checks clearance and grants access to the next member. A prepared flower is held until that access is available; the member can continue reasoning while holding it. Members retain command ownership throughout. Block widths illustrate the coordination pattern rather than measured phase durations.

**Flower-insertion coordination.** The two-flower case uses AOP episode E019, with Direct episode E020 supplying the comparison in Figure 3. The hyper-agent first grants exclusive vase access to the member holding the pink flower, then passes access to the member holding the blue flower. When the blue stem tip is occluded, it pauses insertion and gives the finished member a new responsibility: reposition its wrist camera to reveal the tip. Insertion resumes after the improved view is available, with the camera held in place. The lead then verifies completion. Members retain command ownership throughout: the hyper-agent adapts sensing responsibilities and shared access without taking over direct control.

**Changes in control responsibility.** The lead takes direct control of at least one group in nine of the 31 AOP episodes; the remaining 22 retain delegated control throughout. A control transfer alone does not establish a timing gain. The bottle episode is instructive: despite physical coupling, the lead retains separate members for stabilization and twisting. The choice depends on the task's physical requirements, rather than collaboration always implying direct lead execution.

In the shirt trace, members prepare the lower corners, correct a missed pinch and report grasp readiness. The lead then reclaims both arms for the supported fold. In the first towel trace, control returns to the lead during the first fold and is delegated again for the second. Numerical geometry and visual grasp checks support these local decisions, but their separate causal contribution is not isolated here.

In the second Fixed towel trace, a member reports a quarter-scale footprint around minute 21, followed by continued alignment and curl correction. Direct and AOP also accept residual wrinkles. The same physical rubric must therefore distinguish reaching the requested form from additional finishing. Basket lifting provides a complementary boundary: Direct is faster despite the AOP policy's direct-control option.

**Shared access and partial order.** The toy-block outcome record describes four objects per arm, independent source-side work and exclusive basket access. The towel-pyramid record describes source-side preparation followed by staged placements and updated geometry after lower rolls shift. These traces support the interpretation that synchronization can be localized to a dependency. They do not establish a measured decomposition of savings into inference, motion and waiting.

**Dependencies emerge during execution.** In mixed-object sorting (T06, E075; Figure 5), the lead delegates cubes and batteries to separate members. Transfers shift the containers closer, requiring revised work regions. When the tool assemblies nearly overlap, the lead pauses both pickups and asks the right member to retreat. It verifies clearance, authorizes the left member to place its cubes and withdraw, then resumes the right member. Members retain control throughout. The episode records completion at 9.35 minutes, compared with 14.34 minutes for Direct. This case shows the hyper-agent adapting coordination to dependencies that emerge during initially independent work.

**Interpreting elapsed time.** An agent can be active while computing, waiting or interacting with tools, and holding an object without moving can be essential physical work. The records do not separate these intervals into independently measured inference and motion costs. We therefore report total elapsed time and do not infer aggregate computation cost from apparent activity.

### C.1 Remaining experimental settings

Figures 5–12 show the 19 task/platform settings not displayed in the main paper and the full Direct/AOP comparison for the four-flower teaser, ordered by category and task ID. The same chronological selection rule is used throughout, with every available mode shown, including slower AOP cases and recorded noncompletion. The P-C flower settings include separate two-flower and four-flower configurations; they broaden task coverage without constituting a controlled object-count study. These figures broaden the behavioral record without changing the quantitative denominators.

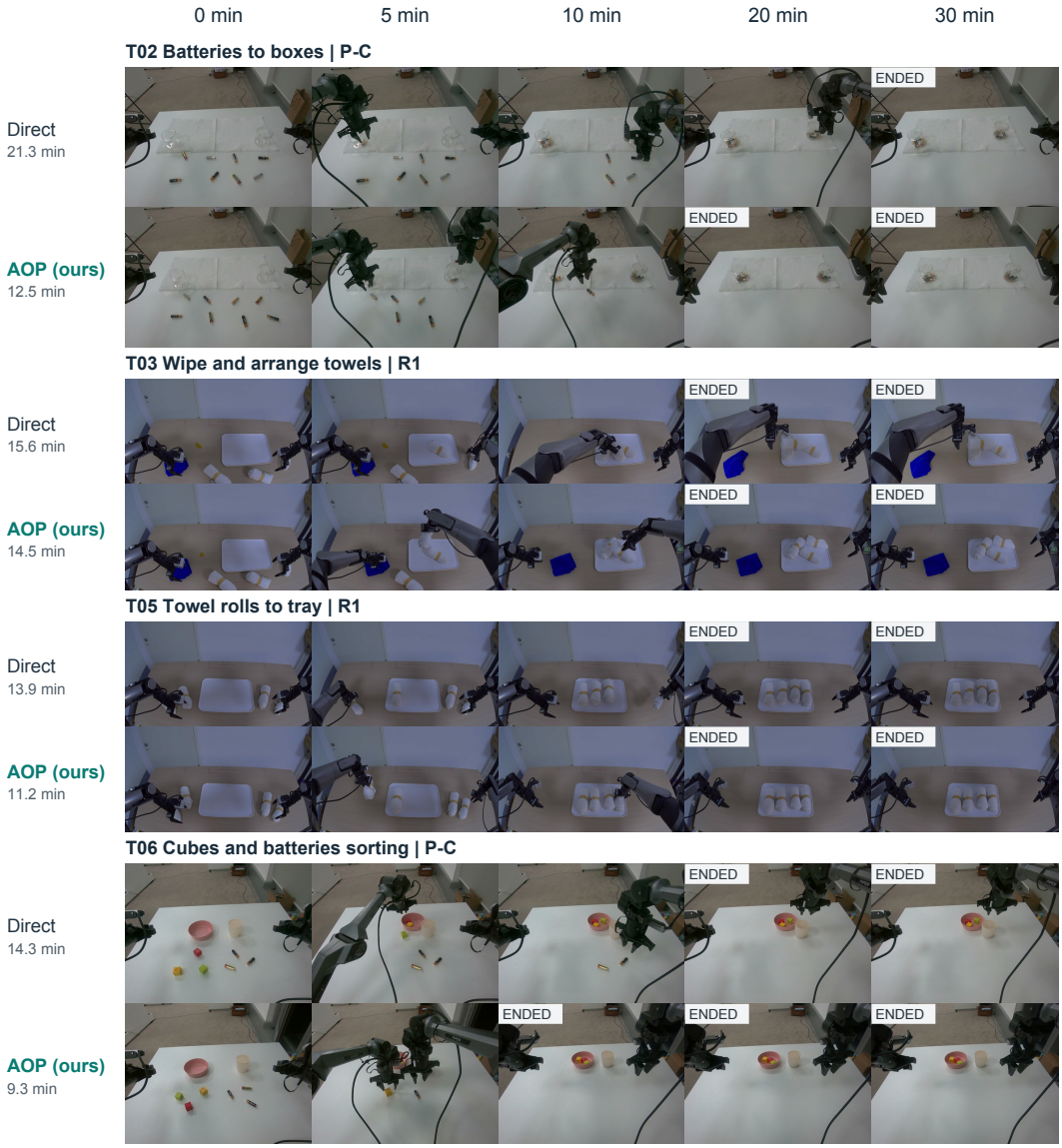



Figure 5: Supplementary sequences: batteries to boxes, wiping and arranging towels, towel rolls to a tray, and independent cubes/batteries sorting. Methods occupy rows; columns use common elapsed times. Labels give observed episode durations. Repeated final images are marked ENDED; they are not later observations.

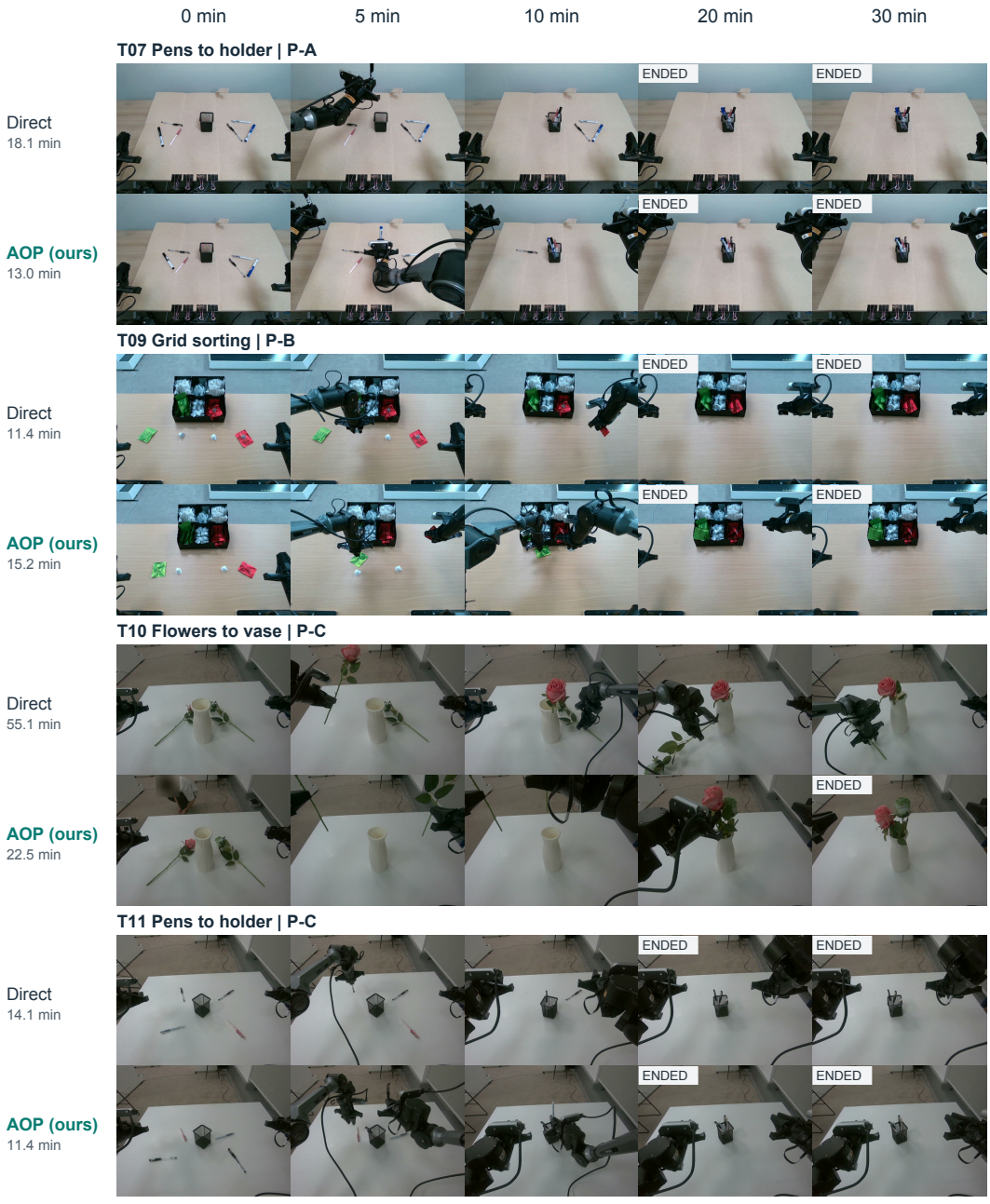



Figure 6: Supplementary sequences: pens on P-A, grid sorting, flowers on P-C, and pens on P-C. Pen placement on P-A is T07 in the task list. Flowers on P-C provide the second installation of the insertion task. Episodes are selected chronologically, without ranking their relative performance.

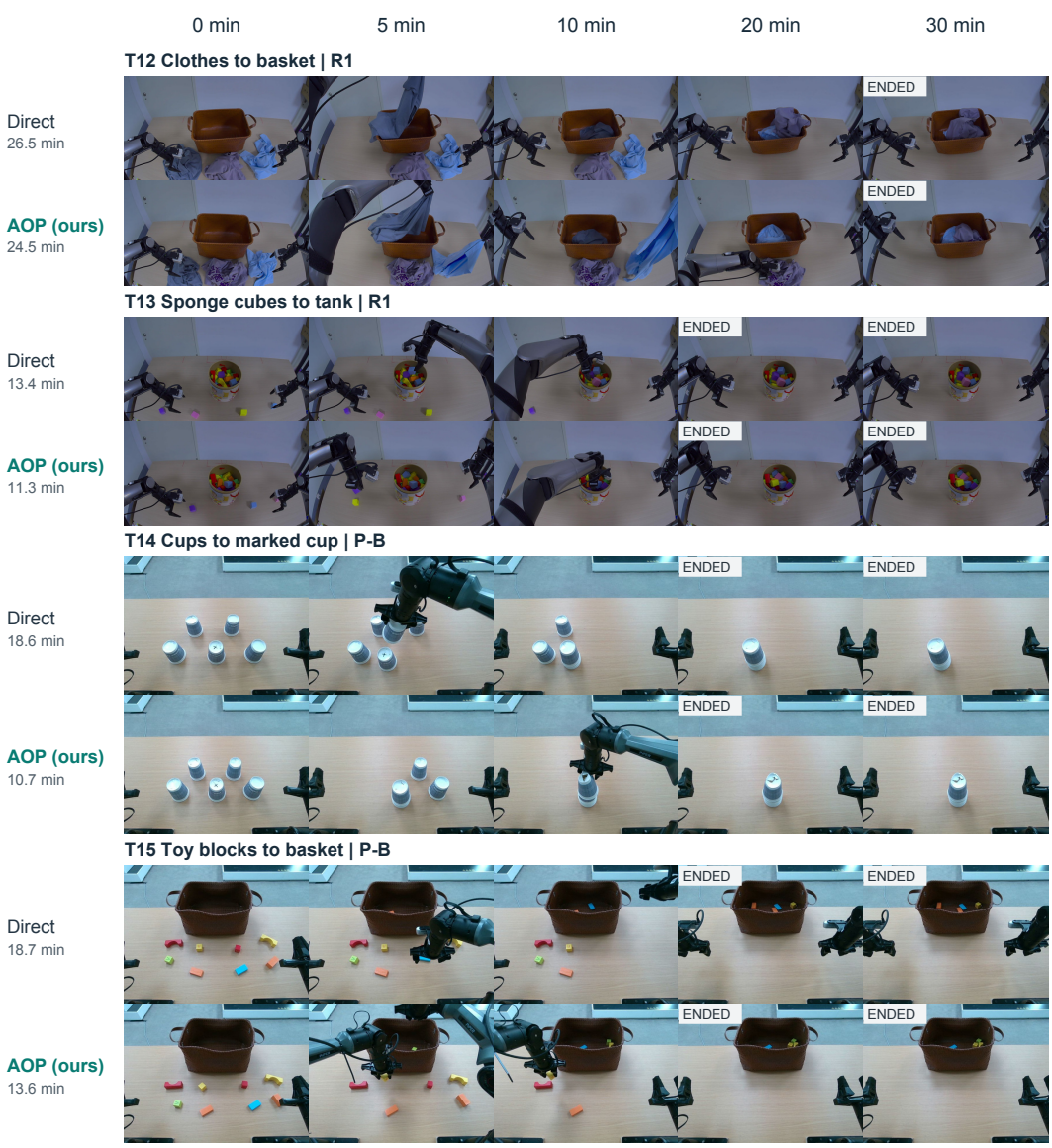



Figure 7: Supplementary sequences: clothes to a basket, sponge cubes to a tank, cups to a marked cup, and toy blocks to a shared basket. Both cup runs declare completion while noting paper-cup damage; their labels retain those recorded outcomes. Initial placements are not asserted identical across methods. All labels use observed time, not the noncompletion penalty.

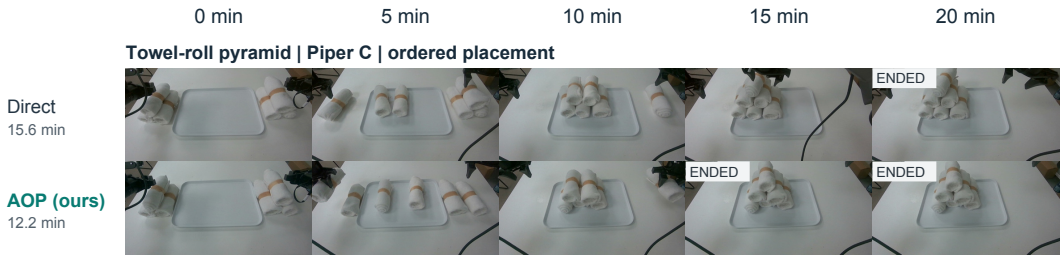



Figure 8: Time-aligned towel-pyramid construction on P-C. Direct and AOP start from their respective recorded scenes. Both complete; AOP combines independent source preparation with ordered tray access, revising placement coordinates when supporting rolls shift. Repeated final images are marked ENDED.

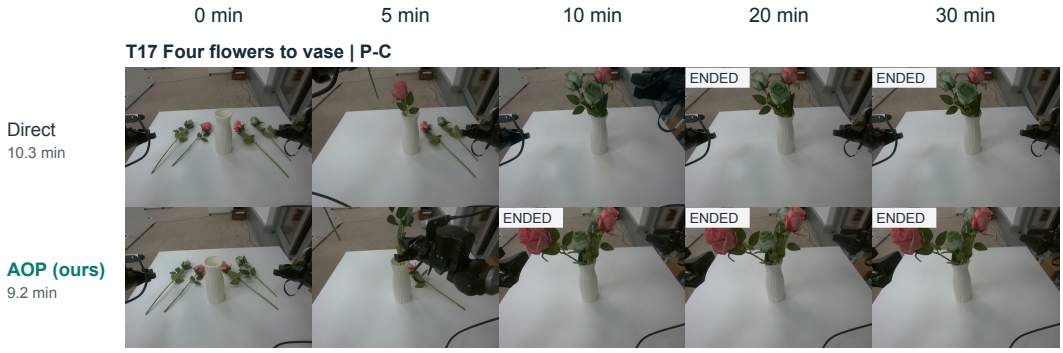



Figure 9: Four-flower insertion on P-C (T17). Both modes record completion: 10.34 minutes for Direct and 9.16 minutes for AOP. The AOP outcome describes parallel source-side work and ordered access to the vase. Each mode recovers from a missed grasp; AOP also corrects a foliage snag, while Direct corrects a stem that initially misses the opening. These are distinct recorded initial scenes, not a controlled comparison with the two-flower setting. Repeated final images are marked ENDED.

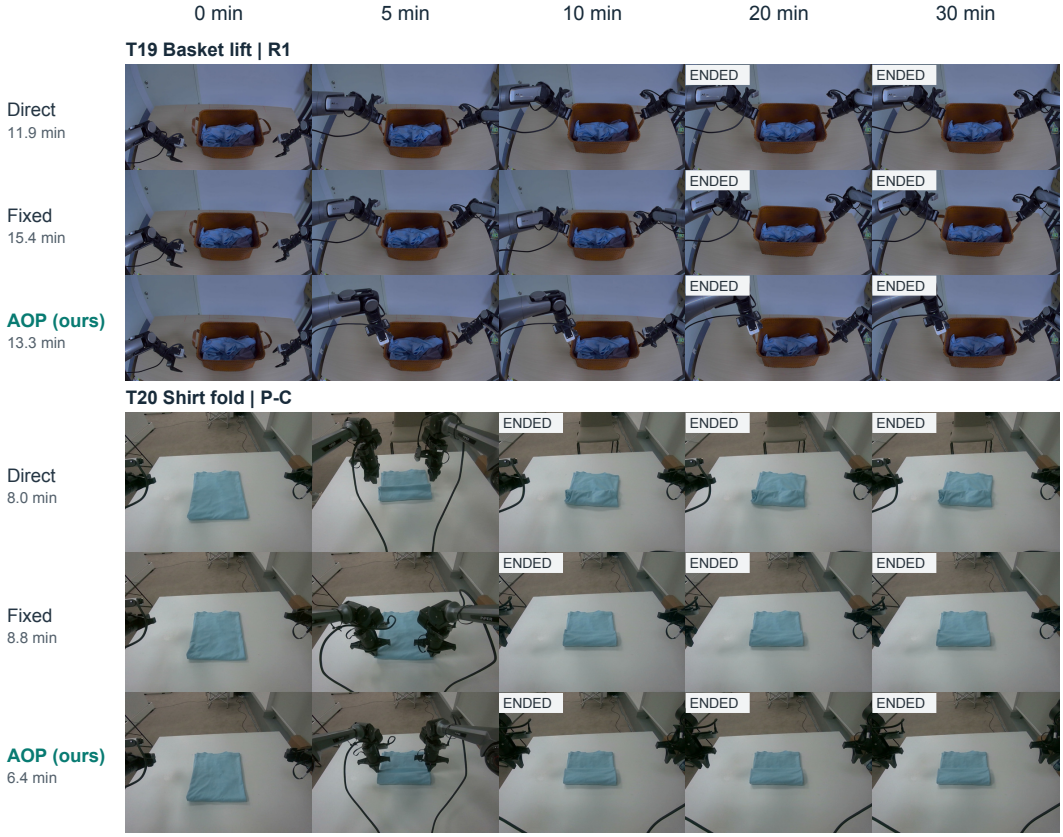



Figure 10: Supplementary Collaborative tasks: basket lifting and shirt folding. Basket lifting includes the faster Direct episode. The displayed Fixed basket run completes, illustrating that permanent member control can also support a shared lift. Repeated final frames are marked ENDED. All available modes are shown under the same chronological selection rule.

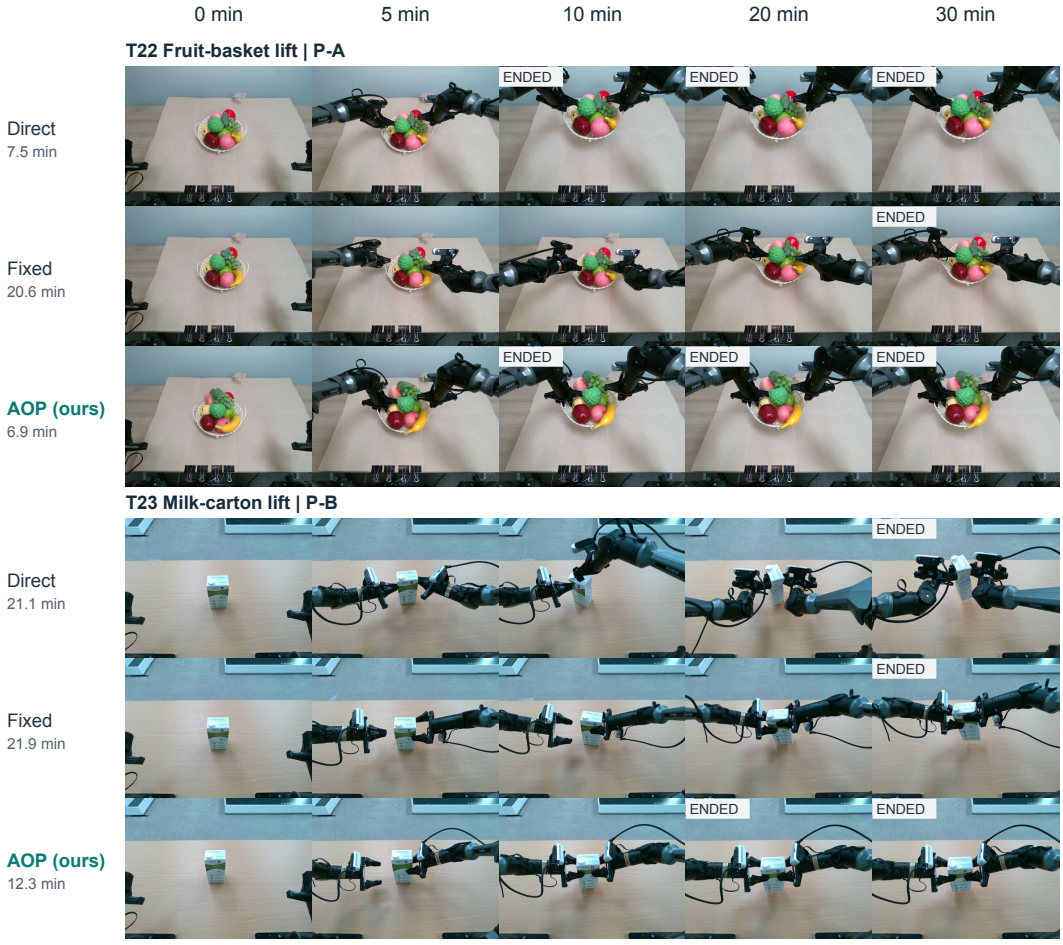



Figure 11: Additional Collaborative settings: fruit-basket lifting on P-A and milk-carton lifting on P-B. All three modes record completion. AOP delegates contact preparation before reclaiming both arms for the shared lift. Camera rows show the earliest recorded episode of each mode; repeated final frames are labeled ENDED. Direct's carton outcome notes creasing without visible leakage. Geometric clearance estimates in these runs are generated by the policies, not an external survey.

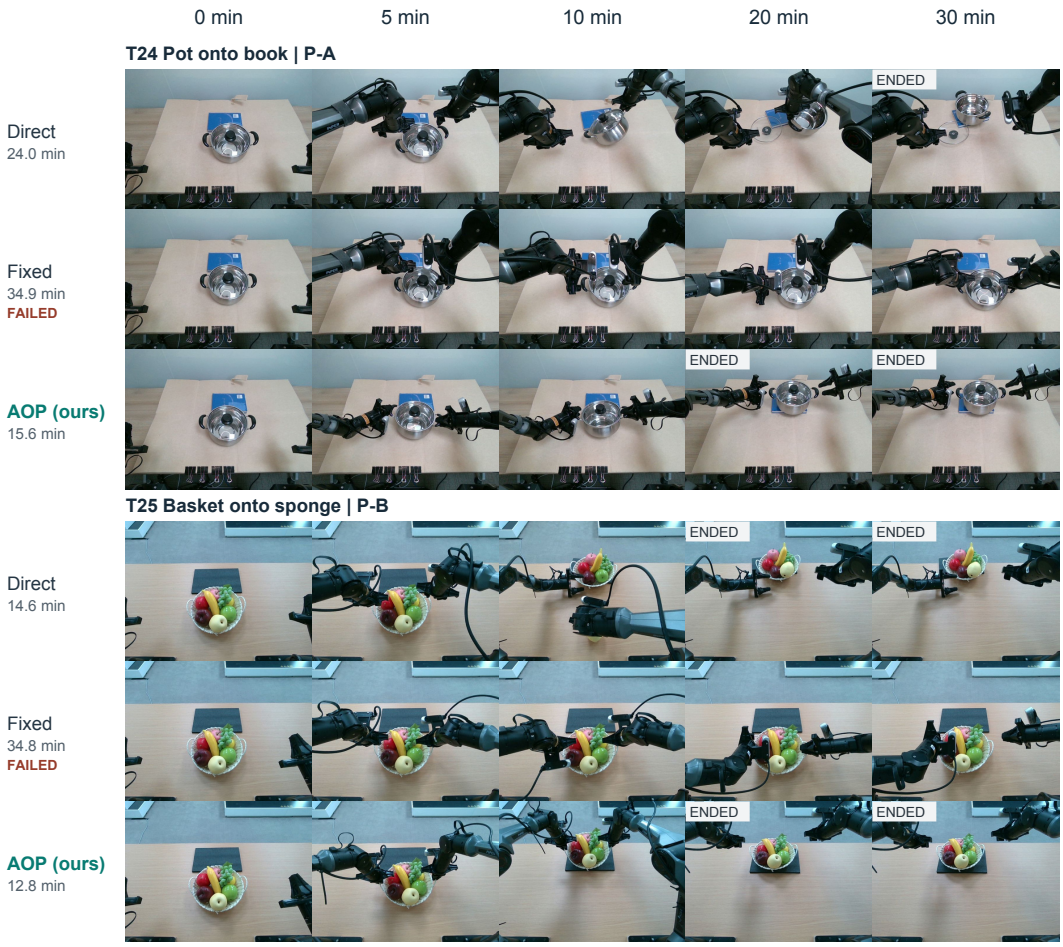



Figure 12: Pot onto a book and fruit basket onto a sponge. Direct and AOP record completion in both settings. Both Fixed runs fail to complete before interruption after 30 minutes. Columns show observations through 30 minutes; row labels mark the final failed outcomes. These traces include contact-model revision, grasp recovery and coordination; photographs alone do not attribute delays to one cause.

## D FIXED DELEGATION: COORDINATION COSTS AND RECOVERY

We analyze all ten Fixed episodes, including successful executions. Under Fixed, members retain control of their groups; the lead can still observe, compute, assign, revise and synchronize. The analysis examines how messages, commands and physical feedback coordinate work across these permanent owners. Fixed constrains who controls each group, while allowing the task schedule to change.

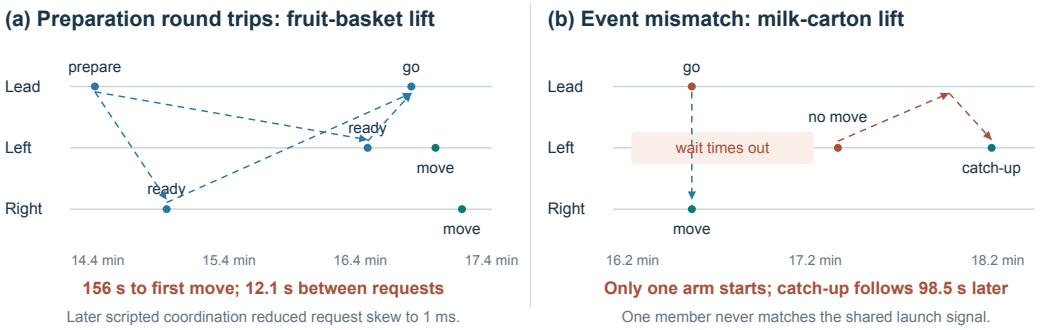



Figure 13: Observed coordination costs under Fixed Delegation. Left: preparation and readiness reports precede movement in the fruit-basket lift. Right: a wait condition does not recognize the shared start signal, delaying one member in the carton lift. Intervals include computation and verification; command timing does not certify motor synchronization. Both episodes eventually complete.

Table 8: Qualitative analysis of all Fixed episodes. Outcomes are recorded labels. Patterns may overlap and do not isolate the causal contributions to elapsed time.

| Episode | Setting / outcome | Observed pattern |
|---|---|---|
| E039 | Towel / stopped | Waypoint revisions precede a coordinated arc; later second-fold grasps repeatedly miss. Contact localization and sequential preparation both contribute. |
| E040 | Towel / stopped | A quarter-sized footprint is reported before continued curl/hem correction. Repeated finishing has little visible effect; completion criteria matter. |
| E045 | R1 basket / completed | Initial hands rise without the basket. Revised engagement and coordinated staged rises recover, showing that fixed ownership can succeed. |
| E046 | R1 basket / stopped | Repeated readiness exchanges, unequal engagement and one-sided diagnostics. Final reports describe staggered progress and basket tilt. |
| E054 | Shirt / completed | Members prepare compatible arcs, advance through lead-approved stages and release successfully. A positive Fixed example. |
| E057 | Bottle / fault | Arm convergence and unverified grasps precede retreat and a reported motor-temperature fault. The bottle topples. End time is unavailable; the 30-minute failure cost is assigned. |
| E061 | P-A basket / completed | Preparation and readiness exchanges precede a staggered lift. A revised synchronization routine reduces the delay between the two motion commands. |
| E064 | Carton / completed | A wait condition does not recognize the shared start signal. One arm lifts while the other times out; a later corrective command brings both to their targets. |

| Episode | Setting / outcome | Observed pattern |
|---|---|---|
| E071 | Pot/book / stopped | Contact-model and feature-identity revisions, repeated grasp attempts and staged test lifts. The transfer to the book is not completed. |
| E074 | Basket/sponge / stopped | Slips and displaced targets trigger alternating exclusive recovery phases, followed by a change to supported sliding. Destination placement remains unfinished. |

**Readiness exchanges add coordination time.** In E061, about 156 seconds elapse between the lead's preparation instruction and the first motion command. This interval includes local kinematic checks, observations and readiness exchanges; it is not an isolated measurement of communication latency. An initial wait also expects a signal while readiness arrives in member reports. The first pair of motion commands is separated by 12.1 seconds. A revised synchronization routine later issues the two commands 1 millisecond apart, and the episode completes. These intervals measure command timing, not physical motor synchronization.

**A shared intention need not produce a shared action.** In E064, both members announce readiness, but the left member's wait condition does not recognize the shared start signal. The right arm begins its lift while the left continues waiting. After the left reports a timeout, the lead authorizes a corrective command, issued 98.5 seconds after the right-arm command. The agent acknowledges the lack of synchronization, and completion follows later. This episode illustrates a recoverable coordination failure within the evaluated policy; it does not establish that all policies with fixed delegation must fail in this way.

**Recovery can invalidate another agent's local plan.** In E074, attempts to establish one contact move the shared basket and change the other member's target. The lead consequently serializes contact refinement so that each member can approach a stationary handle. It later withdraws one arm, grants the other exclusive access, and eventually changes the manipulation strategy. Reports and assignments accompany physical attempts and model revisions. E071 likewise includes substantial geometric troubleshooting. These delays therefore reflect interactions between coordination and changing contact geometry, rather than communication loops alone.

The supported conclusion is that permanent ownership can make coupled preparation, launch and recovery costly. AOP offers another organization: retain local work where useful and let the lead command coupled groups directly when warranted. The Fixed successes, repaired synchronization protocol and imperfect AOP episodes prevent treating that option as a guarantee of optimal timing or reliability.

## E From geometric reasoning to physical plans and execution

The rollout programs suggest a concrete account of how a coding agent handles physical tasks: it constructs a local model for the current goal, computes an executable action under robot constraints, and revises the plan from the object's response. General mathematical knowledge becomes useful through its selection and composition around current observations. A desired relation such as "upright" becomes a geometric constraint; a proposed motion becomes a numerical feasibility question; unexpected feedback becomes a reason to revise the model. This is a hypothesis about adaptability, rather than a causal explanation established by the present evaluation.

We trace four selected episodes from generated code through planning, commands and subsequent observations. Images, depth, calibration, measured state, nominal geometry and generic control/kinematic tools are supplied. The agent chooses task-specific representations, constraints and action parameters. Each analysis links the computed action parameters to issued robot commands and subsequent feedback. These cases explain mechanisms rather than estimate their frequency; numerical validity, actuator tracking and physical outcome remain separate claims.

### E.1 CONVERTING TASK RELATIONS INTO EXECUTABLE ACTIONS

**An object model determines a corrective rotation (E009).** To place a bottle upright during hotel preparation, the right member fits a cylinder to wrist-depth points. The generated program estimates the bottle axis, computes the rotation aligning that axis with vertical, and composes it with the measured tool orientation. The resulting orientation becomes the target of the next motion command. Controller feedback indicates target attainment, followed by observations of placement, release and hand retreat. This connects an object-surface model to an action parameter: the desired bottle relation determines the correction, rather than equating uprightness with a fixed wrist pose. The radial fitting residual does not certify calibration accuracy or stable placement.

**Observed contact geometry revises an approach (E062).** For the milk carton, nominal finger geometry and observed pad locations disagree. A generated program backprojects selected pad pixels and expresses them in the flange frame. The member uses this contact estimate to revise its approach; its recommended 40 mm correction appears as the next $+x$ movement. Lateral insertion and closure then proceed incrementally, with image and gripper-feedback checks. The member reports apparent contact rather than proven load support; a subsequent test lift supplies the next check. Thus code connects the frame controlled by the robot to the physical contact needed by the task, while retaining uncertainty about the installed geometry.

**Numerical constraints expose alternative motion paths (E062).** A straight 80 mm lift at fixed orientation drives the right arm's fourth joint to a bound and leaves substantial pose residual. The agent uses the task's remaining freedom to examine backward-and-upward alternatives with the supplied kinematic solver. Table 9 compares two candidates from the same measured initial flange position, both targeting an approximately 40 mm rise. Moving backward as well as upward yields smaller predicted orientation error. These are candidate computations within one episode, not separate physical trials.

Table 9: Two milk-lift candidates evaluated from the same initial flange position. Values are residuals reported by the supplied kinematic solver, not measured tracking error or collision-clearance bounds. The coding agent chooses and revises the requested waypoints.

| Candidate translation | Position residual (mm) | Rotation residual (rad) |
|---|---|---|
| 40 mm up | 1.654 | 0.04605 |
| 40 mm backward and 40 mm up | 0.208 | 0.00522 |

The later execution makes the plan-to-action link explicit: the computed six-joint target is used in the right-arm command and coordinated with a left-arm target. Controller feedback first indicates motion in progress and then target attainment by both arms. Object-motion checks test whether the carton followed the hands. The executed segment starts from a later state, so its residual must not be compared with the initial 80 mm request as a controlled improvement. The case connects constraint discovery and revised waypoints to command execution and physical verification.

### E.2 USING ACTION TO IMPROVE THE NEXT PLAN

**Thin-stem localization (E023).** Flower-stem depth is unreliable at a thin structure, and the tip's offset from the gripper is uncertain. The agents use front-camera images before and after a measured translation at approximately fixed orientation. In a common arm-base frame, the generated least-squares calculation fits ray scales to the measured displacement:

$$\min_{\lambda} \|\lambda_1 r_1 - \lambda_0 r_0 - \Delta p\|_2^2 . \tag{3}$$

Here $\lambda = (\lambda_0, \lambda_1)$, $r_0, r_1$ are calibrated camera rays and $\Delta p$ is the measured gripper translation. The constraint assumes that the stem moves rigidly with the grasp and that orientation remains approximately fixed. The member and lead compute estimates from different captures and distortion treatments. Their internal residuals are only a few millimeters, yet their tip estimates disagree at centimeter scale. The lead acknowledges the discrepancy and requests a close observation before

descent. Thus a small fitting residual is a consistency check within a model, not a localization-accuracy certificate. The same episode also contains a robust fit of the vase rim, giving an explicit geometric target for alignment.

The raise also supplies clearance for the above-vase approach. The member uses the provisional tip estimate to choose that approach, then refines it from a closer view. The next pose applies the computed correction of $(-24, -7, -45)$ mm, followed by staged descents of 30, 30 and 35 mm. Release and retreat are followed by a check that the flower remains supported. The useful capability is to combine an approximate geometric plan with further sensing and bounded correction; the trace does not show exact open-loop placement.

### E.3 Checking consequences and revising the physical explanation

**Object motion during a shared lift (E059).** For fruit-basket lifting, the lead fits the table plane using iterated SVD with outlier rejection. After staged bimanual lifts, it matches image features, backprojects their depth and projects displacement onto the table normal. The computation tests whether the basket follows the hands rather than treating commanded motion as object motion. One comparison gives 46.04 mm median rise from 36 matched depth pairs; a later comparison gives 94.41 mm from 18 pairs. The latter pairwise interquartile range, 89.17–97.82 mm, describes variation among visible patches, not trial-level uncertainty or the hidden underside's absolute clearance.

In milk-carton lifting (E062), the lead takes direct control of both arms while the members continue their analytical work. Despite the measured gripper rise, the lead initially interprets nearly unchanged depth at selected image locations as evidence of slip. The right member then changes the measurement: it tracks printed carton features with Lucas–Kanade optical flow and backprojects their depth using the supplied camera intrinsics and camera-to-arm transform.

The analysis computes the median coordinates of 23 tracked features in each of three observations (Table 10). Their estimated height increases, contradicting the fixed-pixel diagnosis, and the lead's final assessment acknowledges the correction. The lead has already tightened a gripper and performed a further small lift before receiving this report, so the calculation is evidence of diagnosis revision, not proof that it prevented every unnecessary action.

Table 10: Milk-carton displacement estimated by member-generated code (E062). Height is median feature $z$ in the right-arm base frame, not table clearance or ground-truth object pose. Changes are differences of the reported medians. Only aggregate measurements are available; feature count does not supply an uncertainty interval or additional experimental trials.

| Observation | Median feature height (mm) | Change from first (mm) |
|---|---|---|
| Before lift | 59.16 | 0.00 |
| After first lift | 78.24 | 19.08 |
| After next lift | 89.82 | 30.66 |

The important distinction is between following an image location and following the same material feature. A pixel can sample a different part of the scene after motion; correspondence provides a more relevant test of whether the object followed the hands. Later code fits the table plane to estimate visible lower-surface clearance. The member describes approximately 60 mm visible clearance and a 40 mm practical allowance for edge holes, unseen bottom geometry and registration error. This allowance is not a calibrated confidence bound. A later tracking attempt produces occlusion-related mismatches, which the member declines to use as the primary clearance evidence.

For AOP, the case shows that centralized actuation and distributed analysis can coexist. Reclamation changes who commands the arms while preserving a member's ability to investigate a local physical question. The numerical estimate and the ownership transition therefore concern different responsibilities.

### E.4 What the code suggests about physical adaptability

These cases share a planning-and-execution loop: express the physical aim as a geometric relation, derive an action from observations and constraints, execute it, and inspect the object's response.

Coding makes intermediate models computable and revisable. Familiar operations such as fitting, coordinate transforms, least squares and kinematic search can be composed for different objects and goals during execution. This provides a plausible mechanism for broad task coverage without establishing which capability is the dominant cause of generalization.

The action interface determines what the agent can revise. Harness VLA (Zhang et al., 2026) keeps its manipulation primitives fixed during deployment, leaving the planner to choose their parameters and composition. Our agents can also revise the programs that construct and check actions, as illustrated by the alternative carton-lift paths and feature-based slip diagnosis. Shared calibration, kinematic tools and robot controllers still provide the underlying execution interface. These traces motivate preserving this flexibility; they do not establish a performance advantage over a policy limited to predefined primitives.

The numerical operations are established tools, and comparable reasoning appears in Direct basket lifting (E060) and Fixed towel folding (E039); the latter remains incomplete. Direct milk-carton code (E063) returns a candidate joint configuration with substantial pose residual: producing a numerical result does not ensure that the requested motion is feasible. Computation can therefore expose a modeling or feasibility problem as well as solve one. The proposed role of mathematical reasoning would require interventions and a defined held-out protocol to test causally. For AOP, the relevance is that these analytical and planning responsibilities can be distributed while command responsibilities change with the physical dependencies.

# F Positioning relative to prior control organizations

Table 11 contrasts mechanisms described in the original papers. It is a conceptual comparison, not a reproduction or an empirical ranking. AOP concerns both the asynchronous operation of persistent embodied coding sessions and an agent that coordinates their objectives, dependencies and control responsibilities during execution. The comparison relates these two contributions to prior work on feedback, communication and coordination.

# G Rollout dataset and evaluation materials

We will publicly release the real-world rollout dataset on Hugging Face. The dataset supports study of the observable computational process of embodied coding agents: which physical representation they construct, how they derive actions from it, and how observations and execution feedback change subsequent decisions.

## G.1 Episode contents

Each episode connects its task and policy organization to timestamped observations, generated programs, numerical outputs, agent messages, action requests, robot feedback and recorded outcomes. Available camera and depth observations accompany calibration and measured state needed to interpret the computations. The data link each program to its inputs, numerical results and subsequent actions, and each displayed image to its capture time. Missing observations, failed executions, interventions and assigned failure costs are explicitly identified.

## G.2 Research uses

The analyses in Appendix E illustrate three uses. First, the data permit tracing how goals become computable constraints, such as fitting a bottle's axis before aligning it with gravity. Second, they support studying how action and sensing refine a model: measured motion helps localize a flower tip, and object correspondences can correct a slip diagnosis. Third, messages and command records reveal how analytical work and physical control are divided across agents, including continued member analysis during lead execution. Successful and unsuccessful calculations both matter: a low fitting residual need not imply accurate geometry or successful manipulation. The dataset records observable programs and interaction, rather than unobserved internal model reasoning; causal claims about particular capabilities require further controlled experiments.

Table 11: Closest methodological precedents and the specific relationship to AOP. The descriptions refer to the cited designs, not to claims that these systems could never be extended with other mechanisms.

| Work | Coordination mechanism | Relationship to AOP |
|---|---|---|
| Agent as Policy; CaP-X (Jia et al., 2026; Fu et al., 2026) | Interactive coding, execution and feedback for robot control. | Interactive coding and feedback are shared by our baselines; AOP organizes several persistent policy agents and their control responsibilities. |
| Harness VLA (Zhang et al., 2026) | Memory-guided planning composes a fixed library of analytic primitives and a frozen VLA contact primitive. | Our agents can revise modeling, motion construction and recovery code during execution, as well as their division of control. |
| MALMM (Singh et al., 2025) | Supervisor selects planning, coding and execution using observations and chat history. | Adaptation concerns workflow roles; AOP assigns persistent local control to physical components and can return their authority to the lead. |
| RoCo; EMOS (Mandi et al., 2024; Chen et al., 2025) | Robot dialogue with feedback; embodiment-aware discussion followed by asynchronous local execution. | Our orchestrator changes local objectives and direct control within one coupled embodiment, while member sessions persist. |
| LABOR (Chu et al., 2024) | A language-model coordinator selects uncoordinated and coordinated control stages and skill commands. | AOP distributes ongoing reasoning to persistent component sessions and lets the lead revise their work during execution. |
| DAG-Plan; LLM+MAP (Gao et al., 2024; Chu et al., 2025) | Dependency-graph scheduling or symbolic partial-order planning allocates one- and two-arm work. | AOP lets a coding agent determine assignments and dependencies online and take direct control, without prescribing a task graph. |
| BiCICLe (Palma et al., 2026) | Arm-specific LLMs predict conditioned trajectories; Arms' Debate uses fresh in-context prompts for refinement. | AOP organizes persistent feedback-driven coding sessions whose objectives and command responsibilities can change online. |

### G.3 Evaluation context and reproduction

All modes receive the task instruction, robot-specific operating constraints, hardware descriptions, and access to current images and measured state. A numerical coding workspace provides robot geometry, calibration tools and code execution. Agents can inspect all available cameras; actuator access follows their assigned role. Sessions and task-specific programs start afresh for each rollout.

Direct is instructed to solve the task with whole-robot control. Fixed asks the lead to plan and synchronize permanent control-group members; AOP additionally permits delegation, reassignment and direct lead execution. Members pursue assigned objectives, report supporting observations and coordinate dependencies through the lead. We will release the framework and role instructions alongside the dataset.

The rollouts include task instructions, per-episode outcomes and durations, failure annotations and the analysis needed to reproduce the quantitative results. Recordings retain capture times so that physical progress can be related to coordination decisions.