ECHO: EVOLVING COORDINATION FROM HUMAN OPERATIONS FOR TRAINING-FREE LLM-BASED MULTI-AGENT DECISION MAKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) enable training-free multi-agent decision-making, but human interventions in these systems are typically executed as transient, single-agent commands. This local execution fails to re-coordinate the remaining agents, forcing human operators to repeatedly perform similar manual interventions across episodes. We formulate human-guided multi-agent adaptation as inducing reusable, anchor-conditioned coordination patterns from localized human operations. We introduce ECHO, a training-free framework that grounds each human intervention as an immutable anchor within a shared collaboration graph and dynamically reorganizes the roles, intents, subgoals, and coordination dependencies of responder agents around it without modifying the underlying LLM parameters. The resulting responder-side adjustments are abstracted into identity-agnostic, relational coordination operators that encode transferable team reorganization knowledge while accumulating outcome evidence and human feedback. A structured matching mechanism and an evidence-gated Shadow deployment pipeline progressively promote candidate operators from confirmation-dependent execution to fully autonomous application. Evaluations on SMAC and Overcooked-AI show that ECHO improves immediate complementary coordination and supports cross-episode knowledge reuse. Across continual evolution, ECHO maintains a 100.0% success rate while reducing human intervention rate by 51.8%. Frozen operators also reduce human interaction on held-out tasks, with no harmful automatic applications observed in the evaluated transfer trials. These results show that human operations can provide persistent coordination supervision rather than only momentary control.

Code: `https://anonymous.4open.science/r/-ECHO/`

1 INTRODUCTION

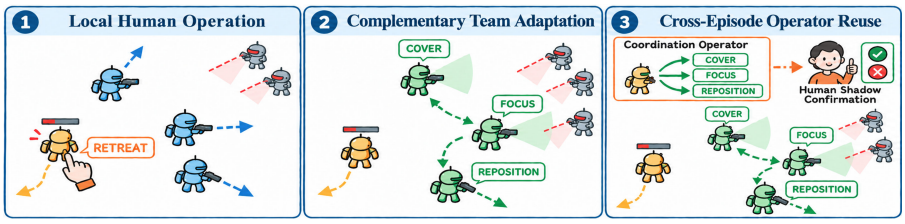

Figure 1: From local human operation to team adaptation and cross-episode operator reuse.

Large language models (LLMs) provide a training-free interface for decision-making through textual observations, task knowledge, and structured action generation Yao et al. (2022); Ahn et al. (2022); Schick et al. (2023). Recent systems extend this capability to multi-agent settings through role assignment, communication, and agent-specific planning Li et al. (2023); Hong et al. (2024). However, dynamic coordination tasks still require human intervention when autonomous decisions become unsuitable. Such interventions are often local, while their intended effects are not. For

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.

example, when a human commands a damaged unit to retreat, its teammates may need to cover the withdrawal, redirect fire, maintain pressure, or clear the retreat path. Executing only the selected agent's command can leave the original team strategy unchanged and produce conflicting behaviors.

This observation exposes a limitation of current human–AI interaction: human operations are typically treated as transient control inputs rather than coordination supervision. Prior work studies teammate adaptation, zero-shot cooperation, and complementary human–AI performance Carroll et al. (2019); Strouse et al. (2021); Yu et al. (2023), yet the team response to an intervention is usually discarded after the current decision or episode. Agent-memory methods retain reflections, trajectories, workflows, or skills across interactions Shinn et al. (2023); Zhao et al. (2024), but these representations describe what was observed or executed rather than how the unselected team should reorganize around a local human intent. The central problem is therefore how to transform a local operation into both an immediate complementary response and reusable coordination knowledge without updating the LLM parameters.

Addressing this problem presents significant challenges beyond storing the operation or the resulting trajectory. (C1) *Complementary coordination without overriding local intent*: executing a local intervention alone leaves team strategy unchanged, whereas unconstrained replanning may overwrite human intent. (C2) *Abstraction of transferable coordination patterns*: commands, trajectories, or skills record what occurred, but do not isolate how the team reorganized for reuse across compatible situations. (C3) *Reliable reuse of accumulated coordination knowledge*: even a matched pattern may be unreliable, requiring execution outcomes and human feedback to determine whether it should be confirmed, rejected, or applied autonomously.

Existing approaches cannot jointly deal with these challenges. Human-in-the-loop and shared-autonomy methods support commands, corrections, or preference feedback, but generally apply them locally or interpret them as global instructions without explicitly representing complementary assignments for the remaining team Bansal et al. (2021); Siu et al. (2021). Conversely, evolving agents reuse textual reflections, episodic experience, executable skills, or optimized workflows Madaan et al. (2023); Wang et al. (2023); Zelikman et al. (2023). These representations do not isolate the structural team transformation associated with a protected intervention, while experience retrieval alone does not determine whether the retrieved knowledge should be confirmed, rejected, or granted autonomous authority Ruan et al. (2024). Thus, existing methods largely address either immediate human control or cross-episode experience reuse, leaving intervention-conditioned team reorganization without an explicit, auditable representation.

We propose **ECHO**, an **E**volving **C**oordination framework derived from **H**uman **O**perations for training-free LLM-based multi-agent decision-making. For (C1), ECHO preserves human operations as immutable anchors while revising the remaining agents to produce complementary coordination. For (C2), it treats the responder-side difference between a state-matched autonomous graph and its anchor-constrained revision as reusable coordination knowledge, abstracting this difference into a relational coordination operator. Unlike a command, trajectory, or skill, the operator encodes a bounded, identity-agnostic team-structure rewrite while leaving action generation to the policy. Relational selectors enable reuse across compatible situations. For (C3), execution and Shadow feedback promote supported operators from candidate to auto-ready, while context ranges, mismatch exclusions, structured matching, ambiguity abstention, human confirmation, and veto regulate their application, thereby separating applicability from autonomous reuse authority.

We evaluate ECHO across SMAC and Overcooked-AI tasks through immediate-coordination comparisons, continual operator evolution, component ablations, frozen-memory transfer, reuse-safety auditing, and subjective-workload assessment. The results show that ECHO propagates local operations into coordinated team responses, reduces repeated human interaction as operator evidence accumulates, and reuses frozen relational operators on held-out tasks without updating the LLM. The main contributions are summarized as:

1. We formulate human-guided multi-agent adaptation as anchor-constrained collaboration-graph reconfiguration, where a local operation induces complementary changes for the remaining team while preserving the grounded human intent as an immutable anchor.
2. We define relational coordination operators as identity-agnostic responder-side graph rewrites, establishing intervention-conditioned team reorganization as a reusable coordination primitive rather than a stored command, trajectory, or skill.

3. We introduce an evidence-gated operator lifecycle that separates knowledge storage from reuse authority, and enable the proposed approach to realize immediate coordination, cross-episode continual evolution, and held-out knowledge transfer.

2 Related Work

2.1 LLM-Based Multi-Agent Coordination with Human Guidance

LLM-based multi-agent systems decompose objectives, assign roles, exchange information, and generate plans through conversational, graph-based, or dynamically organized agent structures Wu et al. (2023); Qian et al. (2024); Liu et al. (2024b); Du et al. (2023); Zhang et al. (2024b); Bo et al. (2024); Zhuge et al. (2024). Cooperative MARL provides complementary foundations for centralized training, value factorization, and decentralized execution Lowe et al. (2017); Rashid et al. (2018); Yu et al. (2022), with SMAC and SMACv2 serving as coordination benchmarks Vinyals et al. (2017); Samvelyan et al. (2019); Ellis et al. (2023). Human–AI studies examine learned teammate models, zero-shot cooperation, and team complementarity Zhang et al. (2024a). Practical LLM-based decision systems and agent benchmarks connect language-based planning with executable multi-agent environments Liu et al. (2024a); Li et al. (2025b;a). Nevertheless, an observed operation is often executed only by the selected agent or treated as an undifferentiated team instruction. ECHO instead fixes the grounded operation as a local anchor and revises the roles, targets, constraints, and relations of the remaining agents. Its contribution is therefore not operation decoding itself, but converting a local operation into a structured complementary team response.

2.2 Experience Reuse and Evolving Agent Systems

Language agents can improve across interactions by retaining reflections, episodic experience, workflows, or reusable skills Park et al. (2023); Wang et al. (2024); Pan et al. (2025); Ouyang et al. (2026); Zhang et al. (2026). Related approaches compose programs, search over reasoning and action trajectories, or optimize agent graphs Zhou et al. (2023), while tool-use research emphasizes executable grounding and deployment risks Yao et al. (2022); Ahn et al. (2022); Schick et al. (2023). These methods reuse textual records, trajectories, or executable skills, which may entangle corrections with environment dynamics and instance-specific entities. ECHO instead derives a bounded relational graph rewrite from an anchor-conditioned responder-side revision, rather than storing the command or episode. Relational selectors enable cross-agent reuse, structured matching controls applicability, and evidence-gated promotion separates storage from authority.

3 ECHO: Evolving Coordination from Human Operations

ECHO bridges within-episode complementary coordination and cross-episode operator evolution, as illustrated in Figure 2. At each decision step t, an LLM collaboration model generates a state-dependent base collaboration graph G_t^0 from the current environmental observation o_t and expert strategy prior P. When a human supervisor intervenes, ECHO grounds the raw input into a protected human anchor (Section 3.1) and re-coordinates responder units to yield an anchor-constrained graph G_t^* (Section 3.2). The difference between G_t^0 and G_t^* is abstracted into a relational coordination operator ω. Over successive episodes, ECHO accumulates positive and negative evidence for each operator and updates its reuse status accordingly, driving an authorization lifecycle (Section 3.3). At later decision steps, stored operators are matched against state observations and applied via evidence-gated Shadow proposals or auto-ready graph rewrites (Section 3.4) to guide downstream policy execution.

3.1 Human Operation Grounding

To decouple interface-specific interaction semantics from high-level reasoning, ECHO employs an environment-specific grounding map D_e. At decision step t, a raw human operation is captured as $u_t = (S_t, m_t, y_t, p_t)$, where S_t denotes the selected unit(s), m_t the operation type, and y_t, p_t optional target entities and spatial coordinates, respectively. The grounding map transforms u_t into a structured human anchor directive $z_t^H = D_e(u_t, o_t) = (S_t, \iota_t^H, \tau_t^H)$, where ι_t^H is the grounded

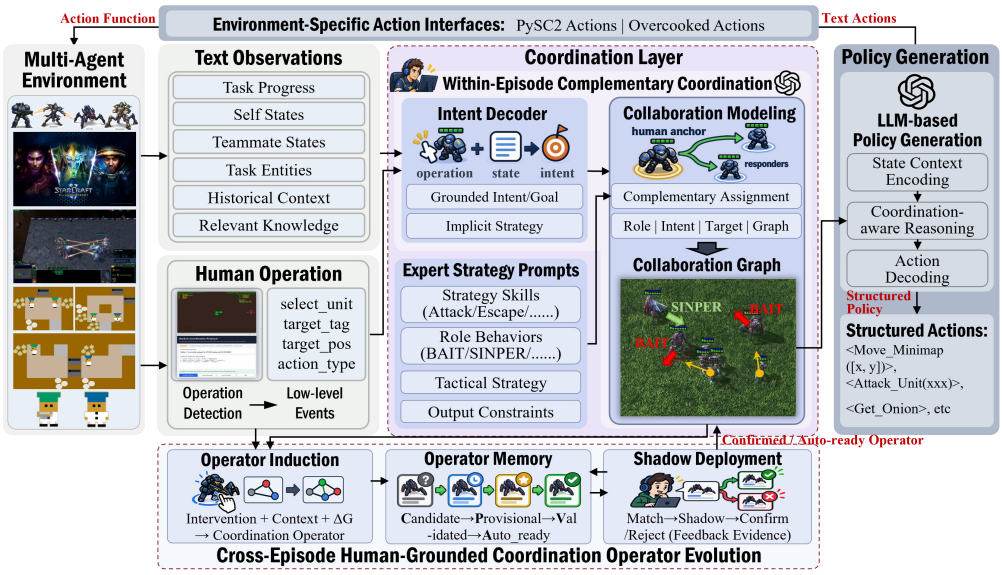

Figure 2: Overview of ECHO. It converts grounded human intent into complementary team coordination and continually induces, validates, and reuses coordination operators through evidence-gated memory and human-auditable Shadow deployment.

tactical intent and τ_t^H is the grounded target specification. For instance, in SMAC, mouse and hotkey operations are normalized according to:

$$\iota_t^H = \begin{cases} \texttt{ATTACK}, & y_t \in \mathcal{E}_t, \\ \texttt{RETREAT}, & d(p_t, \mathcal{E}_t) \geq \bar{d}(S_t, \mathcal{E}_t) + \delta, \\ \texttt{REPOSITION}, & \text{otherwise}, \end{cases}$$

where $\mathcal{E}_t$ denotes the observed enemy entities, $\bar{d}(S_t, \mathcal{E}_t)$ is the average, over the selected units, of each unit's distance to its nearest observed enemy, $d(p_t, \mathcal{E}_t)$ is the distance from the target coordinate p_t to the nearest enemy, and $\delta > 0$ denotes an escape margin. Direct entity clicks assign the unit tag to τ_t^H, whereas terrain clicks assign spatial coordinates. Grounded human intents span $\{\texttt{ATTACK}, \texttt{RETREAT}, \texttt{REPOSITION}\}$, while the collaboration graph extends this vocabulary with `HOLD` as an executable intent and `MAINTAIN` as a persistence label.

The grounded tuple z_t^H establishes a protected human anchor H_t. This preserves the human's intended behavior while giving the subsequent team planning a fixed reference for assigning complementary responses and extracting reusable operator patterns.

3.2 Within-Episode Complementary Coordination

ECHO formulates multi-agent coordination as dynamic reconfiguration over a collaboration graph $G_t = (V_t, E_t)$. Each node $v_t \in V_t$ represents an allied unit or cohesive sub-group, parameterizing its assigned role, tactical intent, target specification, and operational constraints. Directed edge $e_t \in E_t$ encodes relational dependencies, including focus-fire alignment, retreat coverage, mutual support, and interference avoidance.

Given observation o_t and expert strategy prior P (defining role semantics and tactical guidelines), the LLM-based collaboration model generates an unconstrained base graph representing its autonomous strategy hypothesis $G_t^0 = F_{\text{base}}(o_t, P)$. When a human intervention occurs, z_t^H induces a constrained graph reconfiguration $G_t^* = F_{\text{comp}}(G_t^0, z_t^H, o_t, P, \mathcal{O}_t)$, where $\mathcal{O}_t$ represents the current repository of coordination operators learned from human-guided responses. During this step, the anchor nodes' intent and target fields defined by $\mathcal{H}_t$ remain strictly immutable. If a matching operator $\omega \in \mathcal{O}_t$ is retrieved, F_{comp} applies its structural rewrite rules to update responder node parameters and relational edges; otherwise, the LLM zero-shot infers complementary responder assignments conditioned on o_t and the fixed anchor $\mathcal{H}_t$.

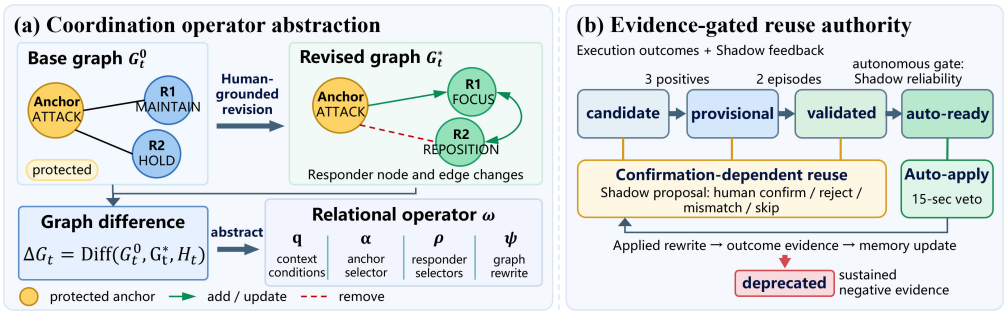

Figure 3: Coordination operator abstraction and evidence-gated reuse in ECHO. (a) Graph changes are abstracted into relational operators; (b) execution and Shadow evidence regulate reuse authority.

To isolate the team's complementary adaptation from the human command itself, ECHO defines the anchor-conditioned responder-side graph revision ΔG_t via a masked differential operator $\Delta G_t = \mathrm{Diff}(G_t^0, G_t^*, \mathcal{H}_t)$. The operator $\mathrm{Diff}(\cdot)$ evaluates node-level attribute changes (role, intent, target, constraints) and edge-level topological modifications (edge additions/deletions) exclusively across non-anchor responder nodes $V_t \setminus S_t$. The revised graph G_t^* directly governs immediate step execution, while ΔG_t serves as the input artifact for operator induction.

3.3 Coordination Operator Induction and Evolution

Relational Operator Abstraction. As in Figure 3(a), rather than recording raw trajectories or specific unit indices, ECHO abstracts the graph differential ΔG_t into an identity-agnostic, relational coordination operator $\omega = (q, \alpha, \rho, \psi)$. q: Context Selector parameterizing environmental prerequisites (e.g., local health thresholds, relative force ratios, spatial proximity ranges). α: Anchor Selector identifying the trigger condition relative to the human anchor (e.g., human_selected_ally, closest_threatening_enemy). ρ: Responder Selector defining relational rules to identify target responders (e.g., closest_support_ally, lowest_health_teammate). ψ: Graph Rewrite Rule specifying bounded modifications to responder roles, intents, subgoals, and coordination edges. Before insertion into memory $\mathcal{O}_t$, episode-specific identifiers are converted into the above relational selectors and rules. Newly induced operators are merged with existing memory entries if their structural signatures overlap above a threshold, updating context ranges and accumulating statistical evidence.

Outcome Evidence & Lifecycle Governance. The validity and efficacy of an applied operator ω are tracked through binary outcome signals e_t^ω:

$$e_t^\omega = \begin{cases} +1, & \text{if } \mathcal{V}_t \wedge \mathcal{C}_t^\omega, \\ -1, & \text{otherwise,} \end{cases}$$

where $\mathcal{V}_t$ verifies graph schema compliance, anchor preservation, and action legal feasibility, while $\mathcal{C}_t^\omega$ evaluates task progress over a fixed execution window ΔT (e.g., enemy health differential in SMAC or recipe stage advancement in Overcooked-AI). Let p_ω, n_ω, and d_ω denote the count of positive outcomes, negative outcomes, and distinct positive episodes, respectively, with negative ratio $r_\omega = \frac{n_\omega}{p_\omega + n_\omega}$. The authorization state $\mathrm{status}(\omega)$ evolves according to a finite-state machine:

$$\mathrm{status}(\omega) = \begin{cases} \text{deprecated}, & \text{if } (p_\omega + n_\omega \geq 5) \wedge (r_\omega > 0.50), \\ \text{auto-ready}, & \text{if } (p_\omega \geq 3) \wedge (d_\omega \geq 2) \wedge (r_\omega \leq 0.15) \wedge \mathcal{A}_\omega, \\ \text{validated}, & \text{if } (p_\omega \geq 3) \wedge (d_\omega \geq 2) \wedge (r_\omega \leq 0.25), \\ \text{provisional}, & \text{if } (p_\omega \geq 3) \wedge (r_\omega \leq 0.25), \\ \text{candidate}, & \text{otherwise.} \end{cases}$$

This lifecycle sets evidence requirements for cross-episode reuse: promotion to validated or auto-ready status requires positive outcomes from multiple episodes and a sufficiently low observed negative-outcome rate. The autonomous gate $\mathcal{A}_\omega$ additionally requires the Shadow-support and response-rejection criteria reported in Appendix Table 25. Overcooked-AI follows the same authority progression with environment-specific evidence thresholds detailed in Appendix B.2.

3.4 Evidence-Gated Shadow Reuse and Policy Execution

As summarized in Figure 3(b), in autonomous decision steps (where no direct human input u_t is provided), ECHO evaluates stored operators $\omega \in \mathcal{O}_t$ for potential application. Each node $i \in V_t^0$ in the base graph is evaluated as a candidate anchor against non-deprecated operators:

$$s_{\omega,i} = \text{Match}(\omega, o_t, G_t^0[i]), \quad \forall \omega \in \mathcal{O}_t,\, i \in V_t^0,$$

where $G_t^0[i]$ denotes candidate anchor i and $s_{\omega,i}$ is its compatibility score of the operator–anchor pair. The matching function enforces hard prerequisites: anchor intent compatibility, semantic target alignment, relational selector instantiation, and context boundary checks. Candidate matches passing a score threshold are ranked lexicographically by authorization status and score. ECHO discards matches below a fixed score threshold and abstains when operator families of equal status have scores within an ambiguity margin. Compatibility determines whether an operator applies; autonomous reuse authority is determined by the evidence lifecycle. Exact scoring and calibration parameters are reported in Appendix B.3.

When an operator ω^* is selected, its application route depends on its lifecycle state:

- **Confirmation-Dependent** (candidate, provisional, validated): Displayed as a visual Shadow proposal overlaying the interface, showing proposed role/intent rewrites. Execution requires explicit human confirmation. Rejection preserves G_t^0 and records negative Shadow feedback, whereas skipping preserves G_t^0 without affecting operator evidence.
- **Autonomous** (auto-ready): Rendered with a temporary 15-second supervisor veto window. If no human override occurs, the graph rewrite ψ is applied automatically to yield G_t^*.

Each operator application is logged, and its execution outcome updates the evidence described in Section 3.3. Finally, the validated collaboration graph G_t^* conditions unit(s)-level policy generation:

$$a_t^i \sim \pi_{\text{LLM}}(o_t, P, G_t^*[i]),$$

where $G_t^*[i]$ contains node i, its relevant neighboring nodes, and their coordination edges. The LLM produces structured textual directives, which are verified against legal environment action spaces and executed via API drivers (PySC2 or Overcooked-AI environment interfaces). Operators thus modify higher-level team coordination topology while delegating low-level control to base policy.

4 Experiments

Our evaluation examines four aspects of ECHO: whether explicit collaboration structure turns local human operations into complementary team responses; whether operator evolution reduces human interaction without degrading performance; whether evolved operators transfer safely to held-out tasks through evidence-gated Shadow deployment; and whether ECHO supports reliable structured-action execution under synchronized inference. The experiments progress from within-episode coordination to cross-episode evolution, cross-task reuse, and system-level deployability.

4.1 Experimental Setup

Benchmarks. We evaluate LLM-PySC2 on nine **SMAC** Li et al. (2025b) micromanagement tasks spanning Easy-to-Hard difficulty levels (`1c3s5z`, `2m_vs_1z`, `2s_vs_1sc`, `2c_vs_64zg`, `2s3z`, `3s5z`, `3s_vs_3z`, `3s_vs_4z`, and `3s_vs_5z`) and three **Overcooked-AI** Carroll et al. (2019) tasks (`cramped_room`, `asymmetric_advantages`, and `coordination_ring`). GPT-5.4 is used as the primary backbone, with all LLM-based methods sharing the same observations, action interface, and decoding configuration within each environment.

Baselines & Variants. *LLM-PySC2* Li et al. (2025b) maps textual observations to executable actions without expert strategy prompts. *RuleExpert* deterministically executes ECHO's handcrafted skills without LLM reasoning, while *HEP (adapted)* Li et al. (2025a) places the same expert strategy knowledge and legal PySC2 action library into a single-stage LLM action generator. *ECHO-Auto* is the two-stage counterpart: it generates a team coordination plan, then decodes the plan into actions without human input or cross-episode memory. *ECHO-Direct* and *ECHO-Static* receive operations

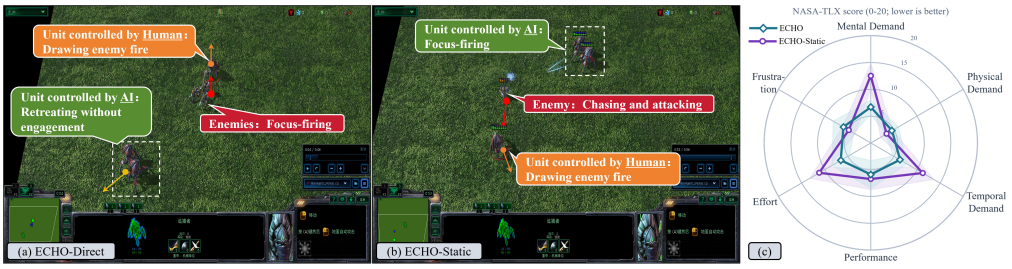

Figure 4: Human-AI behaviors under (a) ECHO-Direct and (b) ECHO-Static (see Appendix C for more), with (c) the mean NASA-TLX profile Hart (2006), details in Appendix A.6.

from the same expert, who follows the same predefined task-specific intervention policy based on ECHO's expert strategy prompts. ECHO-Direct applies each operation only to the selected agents, whereas ECHO-Static propagates it into complementary assignments for the remaining agents; operator memory is disabled in both variants. *ECHO-TextRAG* Shinn et al. (2023) uses natural-language memory of human revisions to augment policy prompts, without structured operators or evidence gating. *ECHO-ConfirmOnly* induces operators and produces Shadow proposals, but requires human confirmation for every application. *ECHO* is the full system, combining complementary coordination, operator induction, evidence-gated evolution, and Shadow deployment. Appendix Table 26 summarizes the matched ablations isolating each ECHO component.

Metrics. We report success rate (SR) and, for SMAC, mean enemy kills/allied deaths; Overcooked-AI additionally uses completion steps. Human interactions (HI) comprise direct operations and Shadow decisions (confirmations, rejections, and skips), reported both jointly and separately. Operator applications (Apps.) count automatic reuse of operators during human interventions or autonomous Shadow deployment. Unless stated otherwise, evaluations use 30 episodes.

4.2 Immediate Complementary Coordination

Table 1: Immediate coordination under matched `gpt-5.4` settings without operator memory. Entries report SR (%) and mean enemy kills/allied deaths. More task data is provided in Appendix A.1.

Method	3s_vs_4z	3s_vs_5z	2c_vs_64zg	2m_vs_1z
LLM-PySC2	0.0 (1.43/3.00)	0.0 (0.70/3.00)	0.0 (21.13/2.00)	0.0 (0.00/2.00)
RuleExpert	40.0 (2.50/1.80)	0.0 (1.57/3.00)	56.7 (47.33/1.17)	6.7 (0.07/1.93)
Full Human Planner	53.3 (3.27/2.13)	6.7 (2.93/2.93)	100.0 (64.00/0.40)	6.7 (0.07/1.93)
HEP (adapted)	30.0 (1.80/1.50)	0.0 (1.63/3.00)	63.3 (45.83/1.03)	10.0 (0.10/1.90)
ECHO-Auto	36.7 (1.83/1.43)	3.3 (1.40/2.97)	80.0 (47.13/0.93)	6.7 (0.07/1.93)
ECHO-Direct (human ✓)	76.7 (3.50/1.00)	16.7 (3.33/2.03)	86.7 (55.37/0.97)	26.7 (0.27/1.73)
ECHO-Static (human ✓)	**100.0** (4.00/0.70)	**30.0** (3.77/1.73)	**100.0** (64.00/0.03)	**66.7** (0.67/1.33)

Table 2: Immediate coordination with 12 participants and no operator memory. Entries report SR (%) and mean enemy kills/allied deaths (#) over 24 trials per map.

Method	3s_vs_3z	3s_vs_4z	2c_vs_64zg	Overall
ECHO-Direct	100.0 (3.00/0.21)	79.2 (3.67/1.54)	75.0 (56.25/0.96)	84.7 (20.97/0.90)
ECHO-Static	**100.0** (3.00/0.04)	**91.7** (3.92/0.71)	**95.8** (63.46/0.46)	**95.8** (23.46/0.40)

To separate immediate complementary coordination from cross-episode evolution, we disable operator memory and evaluate each method over 30 trials per map. ECHO-Direct and ECHO-Static are operated by the same trained expert through the same interface under a predefined task-specific intervention policy. Because complementary propagation changes subsequent states, later interventions remain state-dependent but follow the same policy. Thus, Auto–Direct assesses the value of current human input, while Direct–Static evaluates the cumulative closed-loop effect of complementary propagation. Table 1 shows that LLM-PySC2 fails on all four coordination-intensive tasks.

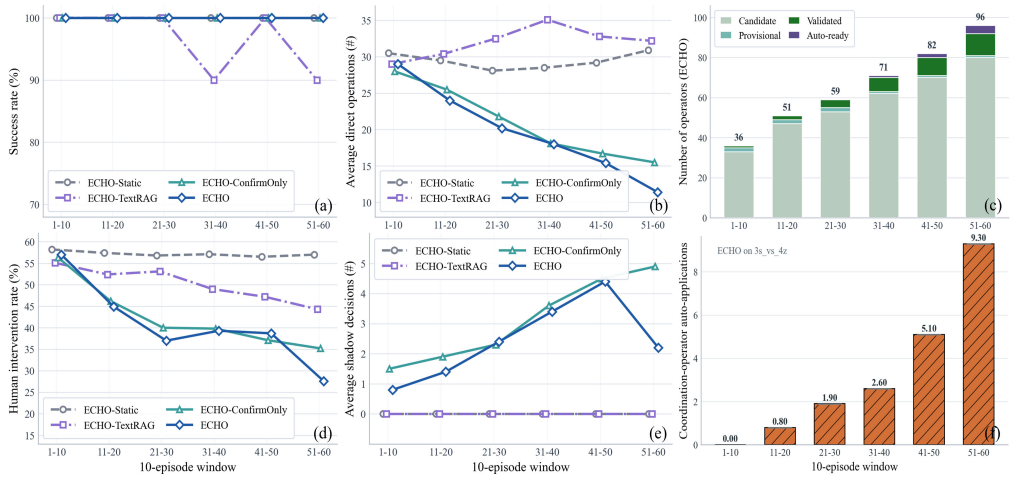

Figure 5: An operator evolution example on SMAC task `3s_vs_4z`: (a) success rate, (b) direct operations, (c) operator composition, (d) human intervention rate, (e) Shadow decisions, and (f) automatic coordination-operator applications across 10-episode windows; details in Appendix A.3.

RuleExpert and HEP (adapted) improve some tasks but remain below ECHO-Static. A matched-frequency Full Human Planner also underperforms ECHO-Static on seven of nine maps (Appendix Table 8), indicating that neither expert knowledge nor direct human control alone explains the gains. ECHO-Direct improves over ECHO-Auto by 6.7–40.0 points, while ECHO-Static further improves over ECHO-Direct by 13.3–40.0 points, reaching 100% on `3s_vs_4z` and `2c_vs_64zg`. Its higher kills and lower allied deaths support the benefit of propagating local operations across the team. Figure 4(a,b) illustrates this effect: ECHO-Direct leaves the AI-controlled unit retreating, whereas ECHO-Static redirects it to focus fire while the human-controlled unit draws enemy attacks.

We further assess immediate complementarity with 12 trained participants in a counterbalanced within-subject study on `3s_vs_3z`, `3s_vs_4z`, and `2c_vs_64zg`. Participants train on separate maps and enter the study after three successful practice trials. Each completes two trials per map under both ECHO-Direct and ECHO-Static, for 12 formal trials in total. The same two scenario seeds are used across conditions for each participant and map. Six participants begin with each condition, and operator memory remains disabled. As shown in Table 2, ECHO-Static increases overall success from 84.7% to 95.8%. Both methods reach 100% on `3s_vs_3z`, but ECHO-Static reduces average allied deaths. Figure 4(c) shows low physical demand and frustration, with mental demand, temporal demand, and effort accounting for most perceived workload. Participant-level results are provided in Appendix A.2.

4.3 Continual Operator Evolution

This experiment tests whether repeated human corrections can be converted into reusable coordination operators. Figure 5 shows one example over 60 consecutive episodes on `3s_vs_4z`, grouped into six 10-episode windows. We report results over three seeds in Appendix A.3.

ECHO preserves a 100% success rate throughout the sequence while reducing total human interactions from 29.8 to 13.6 per episode and direct operations from 29.0 to 11.4. Its human intervention rate accordingly decreases from 57.0% to 27.6%, indicating less overall human involvement rather than a simple change in interaction type. Figure 4(c) similarly shows lower perceived workload for full ECHO than ECHO-Static, mainly in mental demand, temporal demand, and effort. ECHO-ConfirmOnly reduces direct operations from 28.0 to 15.5 but increases Shadow decisions from 1.5 to 4.9, shifting part of the burden from direct control to operator confirmation. In contrast, full ECHO allows sufficiently supported operators to become auto-ready: its Shadow decisions decrease in the final windows while automatic operator applications increase to 9.3 per episode. ECHO-TextRAG does not exhibit the same stable improvement and suffers success-rate drops to 90.0% in two windows, suggesting that unstructured textual experience is less reliable for repeated coordination.

The memory dynamics in Figure 5(c,f) explain this progression. The operator pool expands from 36 to 96 entries, while validated and auto-ready operators gradually emerge. At the same time, automatic coordination-operator applications increase from 0 to 9.3 per episode in the final window, with the sharpest increase occurring in the last two windows. Thus, the growing memory is not merely accumulating records: an increasing fraction of the learned operators becomes reusable at decision time. Taken together, the results show that ECHO converts repeated human corrections into progressively reusable coordination knowledge, reducing human intervention while maintaining task performance on this representative task. Results for additional tasks are in Appendix A.4.

Table 3: Overcooked-AI results after 30 evolution episodes. Entries report SR (%) / completion steps (#) / HI (#) / Apps.(#); detailed statistics are provided in Appendix A.4.

Method	`cramped_room`	`asymmetric_advantages`	`coordination_ring`
RuleExpert	100.0/42.0/0.0/0.0	100.0/39.0/0.0/0.0	100.0/58.0/0.0/0.0
ECHO-Auto	100.0/37.2/0.0/0.0	96.7/37.2/0.0/0.0	100.0/92.7/0.0/0.0
ECHO-Static	100.0/36.5/5.7/0.0	100.0/34.9/8.5/0.0	100.0/47.7/10.7/0.0
ECHO	100.0/**36.3**/**3.1**/1.8	100.0/**33.7**/**5.0**/2.1	100.0/**46.8**/**5.4**/7.5

As shown in Table 3, ECHO maintains a 100% success rate across all three Overcooked-AI tasks while reducing human interactions relative to ECHO-Static. Its operator applications increase to 7.5 per episode on coordination ring, demonstrating effective reuse of learned coordination patterns. ECHO also achieves lower completion steps than ECHO-Auto on all tasks, especially on coordination ring. Representative learned operators are illustrated in Appendix C.6.

4.4 Held-Out Operator Transfer

Table 4: Frozen-memory transfer to held-out tasks. Entries report SR (%) / Apps. (#) / HI (#).

Source Task	Target Task	ECHO-Static	ECHO-TextRAG	ECHO
`3s_vs_4z`	`3s_vs_3z`	100.0/0.0/18.6	93.3/0.0/20.7	**100.0**/**7.5**/**13.5**
`3s_vs_4z`	`3s_vs_5z`	30.0/0.0/47.7	23.3/0.0/49.4	**36.7**/**12.6**/**34.8**
`2s3z`	`3s5z`	26.7/0.0/23.6	33.3/0.0/26.5	**33.3**/**6.1**/**19.7**

We evaluate whether coordination operators learned on one task remain useful on unseen tasks. For each source–target pair, operator memory is induced on the source task, frozen before target-task evaluation, and not updated with held-out evidence.

Table 4 shows that frozen operators transfer to related tasks. On `3s_vs_3z`, ECHO preserves the 100.0% success rate of ECHO-Static while reducing human interactions from 18.6 to 13.5. On `3s_vs_5z`, ECHO improves success from 30.0% to 36.7% and reduces interactions from 47.7 to 34.8. For `2s3z`→`3s5z`, ECHO matches the best success rate while requiring the fewest interactions. Across all targets, ECHO reuses frozen operators on held-out tasks, whereas ECHO-TextRAG provides no operator applications and does not consistently reduce interactions. Appendix Table 15 shows that 62 of 564 Shadow proposals (11.0%) were rejected. Three of 502 confirmed applications were harmful, whereas no harmful outcome was observed among 786 automatic applications.

5 Conclusion

We presented ECHO, a training-free framework that converts local human operations into complementary coordination and reusable cross-episode knowledge. ECHO preserves the human anchor, infers role-aware responder assignments, and stores the resulting relational coordination operators with evidence-gated Shadow reuse. Experiments on SMAC and Overcooked-AI show improved immediate coordination, reduced human intervention as operator evidence accumulates, and evidence-gated transfer to related held-out tasks without updating the LLM. The safety audit found no harmful automatic applications in the evaluated transfer trials, while results on the harder task show that ECHO can maintain comparable success rates while reducing human involvement. These results show that structured coordination memory is a practical and effective mechanism for reliable human-guided adaptation in LLM-based multi-agent systems.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript organization, code debugging, figure design, and LaTeX formatting. They were not used to determine the research questions, make final methodological decisions, conduct the reported experiments, or replace empirical observations. All AI-assisted text was reviewed and revised by the authors. AI-assisted code was inspected and tested against the intended implementation, and all numerical results, claims, and references were independently verified by the authors. The authors take full responsibility for the final content of this work, including all text, code, figures, and claims produced with the assistance of generative AI.

ETHICS STATEMENT

This work evaluates human–AI coordination in simulated SMAC and Overcooked-AI environments and does not involve physical deployment or collection of sensitive personal data. The participant study involved 12 trained adult participants performing simulated coordination tasks. Participants were informed of the study procedure and data collection protocol before participation and provided informed consent. Results are reported only in anonymized and aggregated form. The study involved no foreseeable risk beyond ordinary computer interaction. ECHO retains human supervision during uncertain operator reuse and provides a fixed override period before autonomous application, although the reported safety results should not be interpreted as a guarantee of safety in untested environments.

REPRODUCIBILITY STATEMENT

The method formulation, online decision algorithm, operator representation, evidence criteria, promotion thresholds, matching procedure, and memory-update rules are described in Section 3 and Appendix B. Section 4, Appendix A, and Appendix C specify the environments, baselines, evaluation protocols, random-seed settings, metrics, participant procedure, and complete experimental results. Appendix G provides access to the source code, prompts, fixed experiment configurations, and evaluation scripts needed to reproduce the reported results.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. Is the most accurate ai the best teammate? optimizing ai for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11405–11414, 2021. doi: 10.1609/aaai.v35i13.17359.

Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Benjamin Ellis, Skander Moalla Cook, Vincent Moens, et al. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Advances in neural information processing systems*, volume 36, pp. 51991–52008, 2023.

Zongyuan Li, Chang Lu, Xiaojie Xu, Runnan Qi, Yanan Ni, Lumin Jiang, Xiangbei Liu, Xuebo Zhang, Yongchun Fang, Kuihua Huang, and Xian Guo. Hierarchical expert prompt for large-language-model: An approach defeat elite ai in textstarcraft ii for the first time. *arXiv preprint arXiv:2502.11122*, 2025a.

Zongyuan Li, Yanan Ni, Runnan Qi, Chang Lu, Lumin Jiang, Xiaojie Xu, et al. Llm-pysc2: Starcraft ii learning environment for large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 141492–141554, 2025b.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024a.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024b.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, et al. On memory construction and retrieval for personalized conversational agents. *arXiv preprint arXiv:2502.05589*, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304, 2018.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. In *International Conference on Learning Representations*, volume 2024, pp. 27031–27098, 2024.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2186–2188, 2019.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023.

Ho Chit Siu, Jaime Peña, Edenna Chen, Yutai Zhou, Victor Lopez, Kyle Palko, Kimberlee Chang, and Ross Allen. Evaluation of human-ai teams for learned and rule-based agents in hanabi. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16183–16195, 2021.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14502–14515, 2021.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022.

Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. Learning zero-shot cooperation with humans, assuming humans are biased. In *International Conference on Learning Representations*, 2023.

Eric Zelikman, Qian Huang, Gabriel Poesia, Noah D. Goodman, and Nick Haber. Parsel: Algorithmic reasoning with language models by composing decompositions. In *Advances in Neural Information Processing Systems*, volume 36, pp. 31466–31523, 2023.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: Building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024a.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. Memgen: Weaving generative latent memory for self-evolving agents. In *International Conference on Learning Representations*, volume 2026, pp. 22555–22588, 2026.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132208–132237, 2024b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

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.

A SUPPLEMENTARY EXPERIMENTAL DATA

A.1 COMPLETE RESULTS FOR IMMEDIATE COMPLEMENTARY COORDINATION

Wilson 95% confidence intervals: $[\hat{p}_{\mathrm{L}}, \hat{p}_{\mathrm{U}}] = \frac{\hat{p}+\frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}+\frac{z^2}{4n^2}}}{1+\frac{z^2}{n}}, \hat{p} = x/n, z = 1.96, n = \text{trials}.$

Table 5: Immediate coordination under matched **gpt-5.4** settings. Each cell reports the success rate over 30 trials, with $x/30$ denoting the number of successful trials. Wilson 95% confidence intervals are shown as upper/lower superscripts/subscripts on the success rate. The second line reports average enemy kills/allied deaths. Operator memory is disabled.

Method	**3s_vs_4z**	**3s_vs_5z**	**2c_vs_64zg**
LLM-PySC2	$0.0^{11.4}_{0.0}\%$ (0/30) 1.43/3.00	$0.0^{11.4}_{0.0}\%$ (0/30) 0.70/3.00	$0.0^{11.4}_{0.0}\%$ (0/30) 21.13/2.00
RuleExpert	$40.0^{57.7}_{24.6}\%$ (12/30) 2.50/1.80	$0.0^{11.4}_{0.0}\%$ (0/30) 1.57/3.00	$56.7^{72.6}_{39.2}\%$ (17/30) 47.33/1.17
HEP (adapted)	$30.0^{47.9}_{16.7}\%$ (9/30) 1.80/1.50	$0.0^{11.4}_{0.0}\%$ (0/30) 1.63/3.00	$63.3^{78.1}_{45.5}\%$ (19/30) 45.83/1.03
ECHO-Auto	$36.7^{54.5}_{21.9}\%$ (11/30) 1.83/1.43	$3.3^{16.7}_{0.6}\%$ (1/30) 1.40/2.97	$80.0^{90.5}_{62.7}\%$ (24/30) 47.13/0.93
ECHO-Direct	$76.7^{88.2}_{59.1}\%$ (23/30) 3.50/1.00	$16.7^{33.6}_{7.3}\%$ (5/30) 3.33/2.03	$86.7^{94.7}_{70.3}\%$ (26/30) 55.37/0.97
ECHO-Static	**100.0**$^{100.0}_{88.6}\%$ (30/30) 4.00/0.70	**30.0**$^{47.9}_{16.7}\%$ (9/30) 3.77/1.73	**100.0**$^{100.0}_{88.6}\%$ (30/30) 64.00/0.03

	2m_vs_1z	**1c3s5z**	**2s_vs_1sc**
LLM-PySC2	$0.0^{11.4}_{0.0}\%$ (0/30) 0.00/2.00	$26.7^{44.4}_{14.2}\%$ (8/30) 4.73/7.93	$13.3^{29.7}_{5.3}\%$ (4/30) 0.13/1.87
RuleExpert	$6.7^{21.3}_{1.8}\%$ (2/30) 0.07/1.93	$73.3^{85.8}_{55.6}\%$ (22/30) 8.57/6.43	$66.7^{80.8}_{48.8}\%$ (20/30) 0.67/1.33
HEP (adapted)	$10.0^{25.6}_{3.5}\%$ (3/30) 0.10/1.90	$66.7^{80.8}_{48.8}\%$ (20/30) 8.47/8.03	$70.0^{83.3}_{52.1}\%$ (21/30) 0.70/1.30
ECHO-Auto	$6.7^{21.3}_{1.8}\%$ (2/30) 0.07/1.93	$83.3^{92.7}_{66.4}\%$ (25/30) 8.77/6.80	$66.7^{80.8}_{48.8}\%$ (20/30) 0.67/1.33
ECHO-Direct	$26.7^{44.4}_{14.2}\%$ (8/30) 0.27/1.73	$100.0^{100.0}_{88.6}\%$ (30/30) 9.00/5.17	$100.0^{100.0}_{88.6}\%$ (30/30) 1.00/1.00
ECHO-Static	**66.7**$^{80.8}_{48.8}\%$ (20/30) 0.67/1.33	**100.0**$^{100.0}_{88.6}\%$ (30/30) 9.00/4.33	**100.0**$^{100.0}_{88.6}\%$ (30/30) 1.00/1.00

	3s_vs_3z	**2s3z**	**3s5z**
LLM-PySC2	$0.0^{11.4}_{0.0}\%$ (0/30) 0.97/3.00	$30.0^{47.9}_{16.7}\%$ (9/30) 3.83/4.33	$0.0^{11.4}_{0.0}\%$ (0/30) 2.90/8.00
RuleExpert	$86.7^{94.7}_{70.3}\%$ (26/30) 2.87/2.13	$83.3^{92.7}_{66.4}\%$ (25/30) 4.83/4.00	$0.0^{11.4}_{0.0}\%$ (0/30) 4.20/8.00
HEP (adapted)	$96.7^{99.4}_{83.3}\%$ (29/30) 2.97/2.03	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/3.67	$0.0^{11.4}_{0.0}\%$ (0/30) 3.57/8.00
ECHO-Auto	$93.3^{98.2}_{78.7}\%$ (28/30) 2.93/2.07	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/2.30	$0.0^{11.4}_{0.0}\%$ (0/30) 4.47/8.00
ECHO-Direct	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.30	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/2.87	$16.7^{33.6}_{7.3}\%$ (5/30) 7.17/7.83
ECHO-Static	**100.0**$^{100.0}_{88.6}\%$ (30/30) 3.00/0.07	**100.0**$^{100.0}_{88.6}\%$ (30/30) 5.00/2.83	**26.7**$^{44.4}_{14.2}\%$ (8/30) 7.46/7.73

A.2 Additional Immediate Coordination Results

Table 6: Backbone robustness of **ECHO-Static**. Each cell reports the success rate (%) with its 95% Wilson confidence interval, followed by the number of successful trials over 30 trials and the average enemy kills/allied deaths. All backbones use the same coordination framework, prompts, observations, and action interface without model-specific adaptation.

Backbone	**3s_vs_4z**	**3s_vs_5z**	**2c_vs_64zg**
deepseek-v3.2 (Non-thinking)	$80.0^{90.5}_{62.7}\%$ (24/30) 2.77/1.00	$26.7^{44.4}_{14.2}\%$ (8/30) 3.47/2.83	$100.0^{100.0}_{88.6}\%$ (30/30) 64.00/0.80
qwen3.8-max (Non-thinking)	$83.3^{92.7}_{66.4}\%$ (25/30) 2.90/0.97	$30.0^{47.9}_{16.7}\%$ (9/30) 3.40/2.67	$100.0^{100.0}_{88.6}\%$ (30/30) 64.00/0.70
gpt-4o (Standard)	$83.3^{92.7}_{66.4}\%$ (25/30) 2.83/0.93	$23.3^{40.9}_{11.8}\%$ (7/30) 3.63/2.77	$100.0^{100.0}_{88.6}\%$ (30/30) 64.00/0.73
gpt-5.4 (Reasoning_effort=None)	$100.0^{100.0}_{88.6}\%$ (30/30) 4.00/0.70	$30.0^{47.9}_{16.7}\%$ (9/30) 2.63/1.73	$100.0^{100.0}_{88.6}\%$ (30/30) 64.00/0.03

	2m_vs_1z	**1c3s5z**	**2s_vs_1sc**
deepseek-v3.2 (Non-thinking)	$60.0^{75.4}_{42.3}\%$ (18/30) 0.63/1.40	$100.0^{100.0}_{88.6}\%$ (30/30) 9.00/6.60	$100.0^{100.0}_{88.6}\%$ (30/30) 1.00/1.00
qwen3.8-max (Non-thinking)	$63.3^{78.1}_{45.5}\%$ (19/30) 0.67/1.37	$100.0^{100.0}_{88.6}\%$ (30/30) 9.00/5.90	$100.0^{100.0}_{88.6}\%$ (30/30) 1.00/1.00
gpt-4o (Standard)	$56.7^{72.6}_{39.2}\%$ (17/30) 0.57/1.43	$100.0^{100.0}_{88.6}\%$ (30/30) 9.00/6.87	$100.0^{100.0}_{88.6}\%$ (30/30) 1.00/1.00
gpt-5.4 (Reasoning_effort=None)	$66.7^{80.8}_{48.8}\%$ (20/30) 0.67/1.33	$100.0^{100.0}_{88.6}\%$ (30/30) 9.00/4.33	$100.0^{100.0}_{88.6}\%$ (30/30) 1.00/1.00

	3s_vs_3z	**2s3z**	**3s5z**
deepseek-v3.2 (Non-thinking)	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.13	$96.7^{99.4}_{83.3}\%$ (29/30) 5.00/3.10	$23.3^{40.9}_{11.8}\%$ (7/30) 7.50/7.80
qwen3.8-max (Non-thinking)	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.10	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/2.97	$26.7^{44.4}_{14.2}\%$ (8/30) 7.47/7.63
gpt-4o (Standard)	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.17	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/3.53	$20.0^{37.3}_{9.5}\%$ (6/30) 7.60/7.70
gpt-5.4 (Reasoning_effort=None)	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.07	$100.0^{100.0}_{88.6}\%$ (30/30) 5.00/2.83	$26.7^{44.4}_{14.2}\%$ (8/30) 7.46/7.73

Table 7: Immediate coordination with 12 participants. Each cell reports the success rate (%) over 24 trials per participant-task pair, with the number of successful trials shown as $x/24$. Wilson 95% confidence intervals are shown as upper/lower superscripts/subscripts on the success rate. The second line reports average enemy kills/allied deaths. The Overall column aggregates all 72 participant-task trials. Operator memory is disabled.

Method	**3s_vs_3z**	**3s_vs_4z**	**2c_vs_64zg**	**Overall**
ECHO-Direct	$100.0^{100.0}_{86.2}\%$ (24/24) 3.00/0.21	$79.2^{90.8}_{59.5}\%$ (19/24) 3.67/1.54	$75.0^{88.0}_{55.1}\%$ (18/24) 56.25/0.96	$84.7^{91.2}_{74.7}\%$ (61/72) 20.97/0.90
ECHO-Static	$100.0^{100.0}_{86.2}\%$ (24/24) 3.00/0.04	$91.7^{97.7}_{74.2}\%$ (22/24) 3.92/0.71	$95.8^{99.3}_{79.8}\%$ (23/24) 63.46/0.46	$95.8^{98.6}_{88.5}\%$ (69/72) 23.46/0.40

Table 8: Matched-frequency Full Human Planner results. The human planner uses the same planning frequency and observations as the LLM-based methods and is the same trained expert used in ECHO-Direct and ECHO-Static. Each cell reports the success rate over 30 trials, with $x/30$ denoting the number of successful trials. Wilson 95% confidence intervals are shown as upper/lower superscripts/subscripts on the success rate. The second line reports average enemy kills/allied deaths. Red values indicate success rates lower than those of ECHO-Static on the same map, whereas green values indicate success rates equal to those of ECHO-Static.

Method	**3s_vs_4z**	**3s_vs_5z**	**2c_vs_64zg**
Full Human Planner	$53.3^{69.8}_{36.1}\%$ (16/30) 3.27/2.13	$6.7^{21.3}_{1.8}\%$ (2/30) 2.93/2.93	$100.0^{100.0}_{88.6}\%$ (30/30) 64.00/0.40

	2m_vs_1z	**1c3s5z**	**2s_vs_1sc**
Full Human Planner	$6.7^{21.3}_{1.8}\%$ (2/30) 0.07/1.93	$86.7^{94.7}_{70.3}\%$ (26/30) 8.80/6.13	$46.7^{63.9}_{30.2}\%$ (14/30) 0.93/1.53

	3s_vs_3z	**2s3z**	**3s5z**
Full Human Planner	$100.0^{100.0}_{88.6}\%$ (30/30) 3.00/0.40	$43.3^{60.8}_{27.4}\%$ (13/30) 4.43/4.13	$20.0^{37.3}_{9.5}\%$ (6/30) 6.90/7.50

Summary. The matched-frequency Full Human Planner does not consistently outperform ECHO-Static (in Table 5). Across the nine evaluated maps, it matches ECHO-Static on only two tasks and achieves a lower success rate on the remaining seven. The gap is particularly large on `3s_vs_4z`, `2m_vs_1z`, `2s_vs_1sc`, and `2s3z`. Because the human planner directly controls the selected units at each intervention opportunity without complementary propagation, it requires more direct human involvement while still achieving lower task performance. *These results indicate that ECHO-Static improves coordination efficiency by transforming sparse human operations into complementary team-level assignments, rather than merely replacing LLM decisions with human commands.*

A.3 Complete Results for Continual Operator Evolution

Results are aggregated over **three random seeds**, each comprising 60 episodes. Unless otherwise stated, entries report the mean ± standard deviation across seeds for each 10-episode window. Table 10 instead presents one example seed to illustrate the evolution of operator states.

In addition to the variants defined in Section 4, we evaluate two mechanism ablations. *ECHO-NoSafeguard* selects the highest-scoring legal operator match above the matching threshold and applies its rewrite directly, irrespective of operator status and without Shadow confirmation, context-mismatch exclusion, or ambiguity abstention. Exact ties are resolved by a fixed operator ordering. *ECHO-IDSpecific* replaces relational anchor, responder, and target selectors with episode-specific entity identifiers while retaining the remaining induction, matching, and evolution mechanisms.

Table 9: Continual evolution results on `3s_vs_4z` over six 10-episode windows. Values are reported as mean ± standard deviation.

Method	1–10	11–20	21–30	31–40	41–50	51–60
Success rate (%)						
ECHO-Static	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
ECHO-TextRAG	100.0 ± 0.0	100.0 ± 0.0	96.7 ± 5.8	93.3 ± 5.8	93.3 ± 5.8	93.3 ± 5.8
ECHO-NoSafeguard	96.7 ± 5.8	93.3 ± 5.8	86.7 ± 5.8	83.3 ± 5.8	80.0 ± 10.0	76.7 ± 5.8
ECHO-ConfirmOnly	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	96.7 ± 5.8	100.0 ± 0.0
ECHO	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
Human intervention rate (%)						
ECHO-Static	58.2 ± 0.6	57.5 ± 0.6	56.8 ± 0.6	57.1 ± 0.7	56.5 ± 0.4	57.4 ± 0.4
ECHO-TextRAG	55.2 ± 0.6	53.0 ± 0.6	52.4 ± 0.7	49.0 ± 0.6	46.8 ± 0.4	44.3 ± 0.6
ECHO-NoSafeguard	57.8 ± 0.7	59.6 ± 0.9	61.8 ± 1.1	64.2 ± 1.0	66.7 ± 1.2	68.5 ± 1.3
ECHO-ConfirmOnly	56.3 ± 0.5	46.8 ± 0.6	40.6 ± 0.6	39.8 ± 0.7	37.0 ± 0.2	35.2 ± 0.6
ECHO	57.0 ± 0.6	45.3 ± 0.4	37.7 ± 0.6	39.4 ± 0.7	37.9 ± 0.8	27.5 ± 0.7
Total human interactions (#/episode)						
ECHO-Static	30.5 ± 0.7	29.6 ± 0.5	28.1 ± 0.6	28.6 ± 0.4	29.2 ± 0.3	30.9 ± 0.5
ECHO-TextRAG	29.0 ± 0.4	30.4 ± 0.4	32.4 ± 0.7	34.8 ± 0.3	32.8 ± 0.7	32.1 ± 0.6
ECHO-NoSafeguard	30.8 ± 0.6	33.2 ± 0.8	36.5 ± 0.9	39.4 ± 1.0	41.7 ± 1.1	43.8 ± 1.2
ECHO-ConfirmOnly	29.3 ± 0.4	27.3 ± 0.4	24.1 ± 0.4	21.7 ± 0.3	21.3 ± 0.2	20.3 ± 0.3
ECHO	29.7 ± 0.4	25.4 ± 0.4	22.6 ± 0.3	21.3 ± 0.4	19.7 ± 0.3	13.5 ± 0.3
Direct operations (#/episode)						
ECHO-Static	30.5 ± 0.7	29.6 ± 0.5	28.1 ± 0.6	28.6 ± 0.4	29.2 ± 0.3	30.9 ± 0.5
ECHO-TextRAG	29.0 ± 0.4	30.4 ± 0.4	32.4 ± 0.7	34.8 ± 0.3	32.8 ± 0.7	32.1 ± 0.6
ECHO-NoSafeguard	30.8 ± 0.6	33.2 ± 0.8	36.5 ± 0.9	39.4 ± 1.0	41.7 ± 1.1	43.8 ± 1.2
ECHO-ConfirmOnly	27.9 ± 0.4	25.4 ± 0.5	21.8 ± 0.7	19.4 ± 0.6	16.8 ± 0.3	15.4 ± 0.5
ECHO	28.9 ± 0.5	24.0 ± 0.5	20.2 ± 0.4	17.9 ± 0.6	15.3 ± 0.6	11.3 ± 0.5
Shadow decisions (#/episode)						
ECHO-Static	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ECHO-TextRAG	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ECHO-NoSafeguard	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ECHO-ConfirmOnly	1.4 ± 0.2	1.9 ± 0.1	2.3 ± 0.3	3.6 ± 0.2	4.5 ± 0.4	4.9 ± 0.2
ECHO	0.8 ± 0.1	1.4 ± 0.1	2.4 ± 0.2	3.4 ± 0.3	4.4 ± 0.3	2.2 ± 0.2
Operator auto-applications (#/episode)						
ECHO-NoSafeguard	3.8 ± 0.4	7.1 ± 0.5	10.9 ± 0.7	14.8 ± 0.8	18.2 ± 1.0	21.4 ± 1.2
ECHO	0.0 ± 0.0	0.8 ± 0.1	1.9 ± 0.2	2.6 ± 0.2	5.1 ± 0.3	9.3 ± 0.3

Note: Automatic applications include operators applied during human interventions and autonomous Shadow applications, but exclude human-confirmed Shadow applications.

Table 10: Operator composition over continual evolution on 3s_vs_4z for a single example seed (ECHO). Each entry reports the number of operators (#), with the percentage (%) in parentheses.

Episode Window	Total	Candidate	Provisional	Validated	Auto_ready
1–10	36	33 (91.7%)	2 (5.6%)	1 (2.8%)	0 (0.0%)
11–20	51	47 (92.2%)	2 (3.9%)	2 (3.9%)	0 (0.0%)
21–30	59	53 (89.8%)	2 (3.4%)	4 (6.8%)	0 (0.0%)
31–40	71	62 (87.3%)	1 (1.4%)	7 (9.9%)	1 (1.4%)
41–50	82	70 (85.4%)	1 (1.2%)	9 (11.0%)	2 (2.4%)
51–60	96	80 (83.3%)	1 (1.0%)	11 (11.5%)	4 (4.2%)

Summary. Across three seeds, ECHO maintains a 100.0% success rate while reducing total human interactions from 29.7 ± 0.4 to 13.5 ± 0.3 and direct operations from 28.9 ± 0.5 to 11.3 ± 0.5 per episode. ECHO-ConfirmOnly reduces direct operations, but its Shadow decisions increase from 1.4 ± 0.2 to 4.9 ± 0.2, shifting part of the interaction burden to repeated confirmation. ECHO instead reduces Shadow decisions to 2.2 ± 0.2 as automatic applications reach 9.3 ± 0.3. ECHO-NoSafeguard applies operators more frequently, increasing from 3.8 ± 0.4 to 21.4 ± 1.2, but its success rate falls from $96.7 \pm 5.8\%$ to $76.7 \pm 5.8\%$ and human interactions rise from 30.8 ± 0.6 to 43.8 ± 1.2. This indicates that direct reuse without evidence gating amplifies unreliable coordination changes. ECHO-TextRAG is also less stable, while the single-seed composition example shows the emergence of validated and auto-ready operators. Overall, evidence-gated operator evolution reduces human involvement while preserving reliable task performance.

Table 11: Ablation of relational operator abstraction under frozen-memory transfer.

Source → Target	Method	Source Ops. (#)	SR (%)	HI (#)	Apps. (#)
3s_vs_4z → 3s_vs_3z	ECHO-IDSpecific	132	100.0	23.2 ± 4.5	2.6 ± 1.2
	ECHO	96	100.0	13.5 ± 2.6	7.5 ± 2.4
3s_vs_4z → 3s_vs_5z	ECHO-IDSpecific	132	30.0	49.9 ± 5.7	2.4 ± 1.6
	ECHO	96	36.7	34.8 ± 5.0	12.6 ± 3.4
2s3z → 3s5z	ECHO-IDSpecific	65	33.3	29.1 ± 4.5	1.6 ± 1.1
	ECHO	42	33.3	19.7 ± 3.6	6.1 ± 2.3

Note. Operator memories are induced over 60 source-task episodes and frozen for evaluation on 30 independently target-task episodes, without target-side memory updates. Apps. denotes the number of executed operator applications per episode. Source Ops. counts nondeprecated operators in the source memory after 60 episodes, before frozen-memory transfer.

Summary. ECHO-IDSpecific provides a structured-memory control that retains graph-difference induction, matching, and evidence-gated reuse, but replaces relational selectors with episode-specific identities, thereby isolating the contribution of relational abstraction to cross-task transfer. Across all three transfer pairs, ECHO-IDSpecific retains more source operators but applies fewer of them on the target tasks. ECHO achieves the same or higher success rate with fewer human interactions, suggesting that relational abstraction improves the reuse of frozen operator memory.

A.4 Additional Operator Evolution Evaluation Results

We additionally record continual evolution on 3s_vs_5z over 60 episodes, divided into six 10-episode windows. The first table reports task performance and human interaction metrics for ECHO and ECHO-Static, averaged over three random seeds, while the second records the composition of ECHO's operator memory for one example seed (Table 13).

Table 12: Continual evolution results on 3s_vs_5z over six 10-episode windows.

Method	1–10	11–20	21–30	31–40	41–50	51–60
Success rate (%)						
ECHO-Static	30.0 ± 10.0	26.7 ± 5.8	26.7 ± 5.8	33.3 ± 5.8	23.3 ± 5.8	30.0 ± 10.0
ECHO	26.7 ± 5.8	33.3 ± 5.8	40.0 ± 0.0	36.7 ± 5.8	33.3 ± 5.8	36.7 ± 5.8
Human intervention rate (%)						
ECHO-Static	81.7 ± 3.2	81.4 ± 3.3	85.2 ± 3.0	82.4 ± 5.4	83.0 ± 5.8	85.1 ± 4.3
ECHO	82.4 ± 0.7	80.6 ± 0.9	75.7 ± 1.1	76.8 ± 2.1	70.9 ± 1.1	63.5 ± 0.8
Total human interactions (#/episode)						
ECHO-Static	50.8 ± 1.8	48.4 ± 3.3	48.0 ± 1.2	48.0 ± 1.0	47.9 ± 1.0	48.2 ± 2.5
ECHO	49.7 ± 0.5	48.2 ± 0.4	43.4 ± 0.8	46.8 ± 0.9	41.6 ± 0.7	38.0 ± 0.4
Direct operations (#/episode)						
ECHO-Static	50.8 ± 1.8	48.4 ± 3.3	48.0 ± 1.2	48.0 ± 1.0	47.9 ± 1.0	48.2 ± 2.5
ECHO	46.5 ± 0.6	43.4 ± 0.2	38.1 ± 0.6	39.3 ± 0.6	36.4 ± 0.6	33.9 ± 0.3
Shadow decisions (#/episode)						
ECHO-Static	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ECHO	3.2 ± 0.2	4.8 ± 0.2	5.3 ± 0.3	7.5 ± 0.4	5.2 ± 0.2	4.1 ± 0.1
Operator auto-applications (#/episode)						
ECHO-Static	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ECHO	4.0 ± 0.3	7.5 ± 0.4	6.1 ± 0.3	10.7 ± 0.5	14.1 ± 0.4	16.9 ± 0.4

Table 13: Operator composition over continual evolution on 3s_vs_5z for a single example seed (ECHO). Each entry reports the number of operators (#), with the percentage (%) in parentheses.

Episode Window	Total	Candidate	Provisional	Validated	Auto_ready
1–10	20	20 (100.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
11–20	33	32 (97.0%)	0 (0.0%)	1 (3.0%)	0 (0.0%)
21–30	39	36 (92.3%)	0 (0.0%)	3 (7.7%)	0 (0.0%)
31–40	50	46 (92.0%)	0 (0.0%)	4 (8.0%)	0 (0.0%)
41–50	64	55 (85.9%)	2 (3.1%)	5 (7.8%)	2 (3.1%)
51–60	74	64 (86.5%)	1 (1.4%)	6 (8.1%)	3 (4.0%)

Summary. Despite the higher difficulty of 3s_vs_5z, ECHO progressively reduces human involvement through continual operator evolution. Across three random seeds, ECHO reduces the human intervention rate from 82.4% to 63.5%, total human interactions from 49.7 to 38.0 per episode, and direct operations from 46.5 to 33.9. Its success rate fluctuates between 26.7% and 40.0%, while Shadow decisions peak during the middle windows and later decrease as operator auto-applications increase from 4.0 to 16.9 per episode. Table 13 shows the corresponding maturation of operator memory, with candidate operators becoming less dominant and validated or auto-ready operators emerging over time. Success remains variable on this harder task, but ECHO is more effective at reducing repeated human involvement than at consistently improving task completion under challenging combat conditions.

Table 14: Post-evolution evaluation across SMAC tasks. Each task reports success rate (%) over 30 post-evolution episodes / operator applications per episode (#) / human interactions per episode (#). ECHO is first evolved for 60 consecutive episodes on the same task and then evaluated with the evolved operator memory. ECHO-Static disables operator memory and is evaluated under the same 30-episode protocol.

Map	**ECHO-Static**	**ECHO**	**Reduction** (%)	**Avg. LLM Steps/Ep.** (#)
3s_vs_3z	100.0/0.0/18.6	100.0/8.2/12.9	30.6	53.7
3s_vs_4z	100.0/0.0/29.8	100.0/10.6/13.6	54.4	59.8
3s_vs_5z	30.0/0.0/47.7	33.3/13.5/33.5	29.8	60.4
2s3z	100.0/0.0/9.2	100.0/3.1/6.3	31.5	27.3
3s5z	26.7/0.0/25.0	33.3/8.7/17.6	29.6	42.1

Note. In the ECHO-Static and ECHO columns, each entry is formatted as success rate / operator applications / human interactions. Operator applications count automatic coordination-operator reuse, including operators applied during human intervention and auto-ready Shadow applications, but excluding Shadow applications explicitly confirmed by a human. Human interactions include direct operations and Shadow decisions. **Reduction** is computed from the human-interaction counts as $(H_{\text{Static}} - H_{\text{ECHO}})/H_{\text{Static}} \times 100\%$.

Table 15: Safety audit of frozen-operator reuse on held-out tasks.

Source → Target	Proposals	Confirmed / Rejected	Auto Apps.	Harmful Confirmed / Auto
3s_vs_4z → 3s_vs_3z	147	136 (92.5%) / 11 (7.5%)	225	1/136 (0.7%) / 0/225 (0.0%)
3s_vs_4z → 3s_vs_5z	196	177 (90.3%) / 19 (9.7%)	378	2/177 (1.1%) / 0/378 (0.0%)
2s3z → 3s5z	221	189 (85.5%) / 32 (14.5%)	183	0/189 (0.0%) / 0/183 (0.0%)

Note. Confirmed and rejected rates are relative to Shadow proposals. Harmful rates are relative to confirmed and automatic applications, respectively. An application is harmful if its executed rewrite violates the protected anchor or produces an adverse intent-specific outcome. No harmful automatic application was observed among 786 executions (0.0%; Wilson 95% CI: 0.0–0.5%).

Table 16: Detailed Overcooked-AI evaluation statistics over 30 trials per method–task combination. Completion steps, human interactions, and operator applications are reported as mean ± standard deviation.

Layout	**Method**	**Success**	**Completion Steps**	**Human Interactions**	**Operator Applications**
cramped_room	ECHO-Auto	30/30	37.23 ± 1.04	0.0 ± 0.0	0.0 ± 0.0
	ECHO-Static	30/30	36.53 ± 0.64	5.67 ± 3.37	0.0 ± 0.0
	ECHO	30/30	36.33 ± 0.92	3.13 ± 0.82	1.80 ± 0.76
asymmetric_advantages	ECHO-Auto	29/30	37.20 ± 2.35	0.0 ± 0.0	0.0 ± 0.0
	ECHO-Static	30/30	34.87 ± 2.67	8.53 ± 5.29	0.0 ± 0.0
	ECHO	30/30	33.67 ± 1.52	5.00 ± 1.66	2.07 ± 0.98
coordination_ring	ECHO-Auto	30/30	92.73 ± 27.30	0.0 ± 0.0	0.0 ± 0.0
	ECHO-Static	30/30	47.67 ± 1.68	10.67 ± 4.05	0.0 ± 0.0
	ECHO	30/30	46.83 ± 0.70	5.40 ± 0.86	7.50 ± 3.19

A.5 Reliability and Efficiency Details

We audit ECHO's LLM outputs at the response level to separate structural reliability from tactical optimality. Among 1,803 responses, 31 are clearly incorrect (1.72%), including 26 unparsable outputs and 5 nonexistent-target references, giving a strict hallucination rate of 0.28%. Another 88 responses (4.88%) become invalid only after generation because the target dies or leaves the observable region before execution; these are treated as operational anomalies rather than hallucinations. A balanced input-length test with 300 responses per selected map shows no monotonic increase in clearly incorrect responses from approximately 2k to 7k input tokens, suggesting that scene complexity is more important than context length alone. Expert interventions further indicate that the remaining efficiency bottleneck is tactical refinement rather than output validity: interventions range from 0.0 to 37.7 per episode across the nine maps and are concentrated on coordination-intensive tasks such as `2m_vs_1z`, `3s5z`, and `3s_vs_5z`.

Table 17: Response-level reliability audit of ECHO's LLM outputs.

Category	Count	Rate
Total responses	1803	100.00%
Broader operational anomalies	119	6.60%
Clearly incorrect	31	1.72%
Unparsable	26	1.44%
Strict hallucinations	5	0.28%
Post-generation invalidations	88	4.88%

Note. The categories follow a hierarchical structure: *broader operational anomalies* consist of *clearly incorrect responses* and *post-generation invalidations* ($119 = 31 + 88$), while *clearly incorrect responses* consist of *unparsable outputs* and *strict hallucinations* ($31 = 26 + 5$). *Strict hallucinations* refer to nonexistent-target responses. *Post-generation invalidations* are initially valid actions that become invalid before execution because the target dies or leaves the observable region. All rates are calculated over the 1,803 responses; parent and child categories should not be summed.

Table 18: Input-length sensitivity. Each map contains 300 audited LLM responses under the same model (gpt-5.4), decoding settings, and output constraints.

Map	Approx. Tokens	Incorrect	Rate
`2m_vs_1z`	2k	2/300	0.67%
`3s_vs_5z`	4k	0/300	0.00%
`3s5z`	7k	3/300	1.00%
`2c_vs_64zg`	7k	10/300	3.33%

Note. Incorrect follows the definition of clearly incorrect responses in Table 17. The results are descriptive: error rates do not vary monotonically with approximate input length, while the two 7k-token maps exhibit different rates, suggesting that input length alone does not determine output reliability and that task-specific grounding difficulty may also contribute.

Table 19: Common LLM inference settings. Backbone-specific models and inference modes are reported in Table 6.

Parameter	Setting	Description
Temperature	0.1	Sampling temperature
n	1	Responses per query
Maximum actions per query	1	Actions per unit/group
Timeout	15 s	API wait limit
Retry limit	5	Failed-call retries
Prompting	Fixed and structured	Shared prompt template
Response format	Structured text	Executable action strings
Query protocol	Synchronized	Paused during inference

A.6 HUMAN-STUDY WORKLOAD AND EVALUATION PROTOCOLS

Table 20: NASA-TLX scores of the 12 participants in the immediate-coordination user study. Operator evolution and Shadow suggestions are disabled, and participants evaluate ECHO-Static. Values are reported per participant, with the mean and standard deviation (SD) in the last row.

Participant	Mental	Physical	Temporal	Performance	Effort	Frustration	Overall
P1	18	2	7	10	6	5	8.00
P2	14	3	8	4	11	4	7.33
P3	13	5	9	8	12	3	8.33
P4	13	3	10	7	12	5	8.33
P5	11	5	13	4	9	3	7.50
P6	13	2	13	8	11	5	8.67
P7	12	6	11	6	12	7	9.00
P8	11	2	11	9	11	6	8.33
P9	9	3	12	4	12	6	7.67
P10	12	3	15	7	11	3	8.50
P11	14	3	13	7	13	6	9.33
P12	10	4	13	7	14	4	8.67
Mean $\pm$ SD	12.50 ± 2.32	3.42 ± 1.31	11.25 ± 2.38	6.75 ± 1.96	11.17 ± 2.04	4.75 ± 1.36	8.31 ± 0.60

Table 21: NASA-TLX scores of the 12 participants with operator evolution and Shadow suggestions enabled and operator memory obtained after 60 evolution episodes. Participants evaluate the full ECHO system. Values are reported per participant, with the mean and standard deviation (SD) shown in the last row.

Participant	Mental	Physical	Temporal	Performance	Effort	Frustration	Overall
P1	6	2	5	6	7	3	4.83
P2	6	5	8	10	9	6	7.33
P3	8	5	5	9	5	5	6.17
P4	8	5	8	5	6	5	6.17
P5	9	6	7	6	7	6	6.83
P6	6	3	6	3	3	9	5.00
P7	6	5	9	1	9	10	6.67
P8	7	3	6	4	8	3	5.17
P9	5	5	5	7	5	4	5.17
P10	6	5	6	5	7	5	5.67
P11	8	7	4	9	4	8	6.67
P12	5	4	7	6	8	7	6.17
Mean $\pm$ SD	6.67 ± 1.30	4.58 ± 1.38	6.33 ± 1.50	5.92 ± 2.61	6.50 ± 1.93	5.92 ± 2.23	5.99 ± 0.81

Evaluation Protocols. Immediate-coordination methods are evaluated over 30 episodes per map using matched scenario seeds and, when available, pre-intervention states. The same expert operates ECHO-Direct and ECHO-Static under identical instructions, displays, and interfaces, following a predefined intervention policy derived from the task-specific expert strategy prompts. This policy governs when to intervene, which units to select, and which operation and target to use. ECHO-Direct applies each operation only to the selected units, whereas ECHO-Static additionally revises responder assignments through complementary coordination. Because this revision changes the subsequent trajectory, replaying identical commands after divergence could produce state-inappropriate interventions. Subsequent operations therefore remain state-dependent but follow the same policy in both variants. This policy-matched protocol evaluates the cumulative closed-loop effect of complementary coordination over complete episodes. Operator memory is disabled in both variants.

The participant study uses a counterbalanced within-subject design with 12 trained participants and matched scenario seeds. Each participant completes two trials per map under ECHO-Direct and ECHO-Static. For workload assessment, the same participants also evaluate full ECHO initialized with operator memory obtained after 60 evolution episodes. Operator evolution and Shadow sugges-

tions are disabled for ECHO-Static but remain enabled for full ECHO, with NASA-TLX collected after each condition.

Operator-evolution results use three 60-episode sequences with distinct random seeds. Memory is initialized empty per sequence and retained across episodes, with results grouped into six 10-episode windows. Transfer uses frozen source-task memories over 30 target episodes without target-side updates. Non-autonomous matches remain subject to Shadow confirmation, rejection, or skipping.

All matched comparisons use the same backbone, prompts, observations, decoding settings, and action interface unless stated otherwise.

NASA-TLX questionnaire. After each condition, participants completed the unweighted questionnaire reproduced below.

NASA-TLX Subjective Workload Questionnaire

Instructions.
Rate each dimension using an integer from 0 to 21 based on the task you have just completed.

Dimension	Question
Mental Demand	How much thinking, deciding, remembering, calculating, or perceiving was required?
Physical Demand	How much physical activity, effort, manipulation, or movement was required?
Temporal Demand	How much time pressure or urgency did you experience during the task?
Performance	How satisfied were you with your task performance?
Effort	How much effort was required to achieve your level of task performance?
Frustration	How insecure, discouraged, irritated, stressed, or frustrated did you feel?

Scale.
For all workload dimensions, $0 =$ *Low* and $21 =$ *High*. For Performance, $0 =$ *Very satisfied* and $21 =$ *Very dissatisfied*.

B Supplementary Method Details

B.1 Online Decision Algorithm

Algorithm 1 ECHO Online Decision Loop

Require: Observation o_t, expert prior P, operator memory $\mathcal{O}_t$, optional human operation u_t

$G_t^0 \leftarrow F_{\text{base}}(o_t, P)$
if u_t is available **then**
  $z_t^H \leftarrow D_e(u_t, o_t)$
  $G_t^* \leftarrow F_{\text{comp}}(G_t^0, z_t^H, o_t, P, \mathcal{O}_t)$
  $\Delta G_t \leftarrow \text{Diff}(G_t^0, G_t^*, \mathcal{H}_t)$
  if $\Delta G_t \neq \emptyset$ **then**
    Induce and merge the resulting operator into $\mathcal{O}_t$
  end if
else
  Evaluate $s_{\omega,i}$ for each nondeprecated $\omega \in \mathcal{O}_t$ and $i \in V_t^0$
  Select $\hat{\omega}$ using compatibility, status ranking, and ambiguity abstention
  if no operator passes the reuse gates **then**
    $G_t^* \leftarrow G_t^0$
  else if $\hat{\omega}$ is *auto-ready* **then**
    Display $\hat{\omega}$ with a veto window
    if not overridden **then**
      $G_t^* \leftarrow \hat{\omega}(G_t^0)$
    else
      $G_t^* \leftarrow G_t^0$
    end if
  else
    Present $\hat{\omega}$ as a Shadow proposal
    if confirmed and the anchor remains valid **then**
      $G_t^* \leftarrow \hat{\omega}(G_t^0)$
    else
      $G_t^* \leftarrow G_t^0$
    end if
  end if
end if
for each agent i **do**
  $a_t^i \leftarrow \pi_{\text{LLM}}(o_t, P, G_t^*[i])$
end for
Validate, decode, and execute $\{a_t^i\}$ through the environment interface
Update $\mathcal{O}_{t+1}$ using Shadow feedback and evaluated execution outcomes

B.2 Overcooked-AI Instantiation

ECHO retains the same operator lifecycle across environments while adapting its grounding and evidence definitions to environment-specific interaction semantics. In Overcooked-AI, a human operation selects one chef and an anchor intent from

$$\mathcal{I}_{\text{OC}} = \{\texttt{GET_ONION}, \texttt{DELIVER_ONION}, \texttt{START_COOKING}, \texttt{GET_DISH}, \texttt{PICK_SOUP}, \texttt{SERVE_SOUP}, \texttt{WAIT_NEAR_POT}, \texttt{CLEAR_PATH}\}. \tag{1}$$

The selected intent is fixed as the anchor assignment. The collaboration model assigns the other chef a legal complementary intent conditioned on its held object, the pot phase, and the available stations.

Each operator is indexed by its anchor intent, pot phase, and relational responder selector. A human-grounded revision provides positive evidence when it changes the responder assignment relative to the base graph; an unchanged assignment provides no evidence. Confirmed and auto-ready reuse increments the application count. A context mismatch or response rejection increments the rejection count, whereas a skipped proposal does not affect operator evidence.

Let p_ω and n_ω denote the positive and rejection counts, with $r_\omega = n_\omega/(p_\omega + n_\omega)$ when evidence is available. The environment-specific promotion rule is

$$\mathrm{status}_{\mathrm{OC}}(\omega) = \begin{cases} \textit{auto-ready}, & p_\omega \geq 5,\ r_\omega \leq 0.15, \\ \textit{validated}, & p_\omega \geq 3,\ r_\omega \leq 0.25, \\ \textit{provisional}, & p_\omega \geq 2, \\ \textit{candidate}, & \text{otherwise.} \end{cases} \tag{2}$$

Thus, SMAC and Overcooked-AI share the same evidence-gated authority progression, while their evidence sources and promotion thresholds reflect their respective interaction semantics.

B.3 Operator Matching and Memory Update

For Shadow reuse, ECHO evaluates every nondeprecated operator $\omega \in \mathcal{O}_t$ against each potential anchor i in the base graph. Anchor intent and semantic target class are first checked for compatibility, after which the relational responder selectors are instantiated. Let $\mathcal{N}_\omega$ and $\mathcal{C}_\omega$ denote the numerical and categorical conditions in q. The compatibility score is

$$s_{\omega,i} = \frac{1 + \sum_{k \in \mathcal{N}_\omega} c_k + \sum_{k \in \mathcal{C}_\omega} \mathbb{I}[x_k = \hat{x}_k]}{1 + |\mathcal{N}_\omega| + |\mathcal{C}_\omega|}, \tag{3}$$

where x_k is the observed feature and $\hat{x}_k$ is the stored categorical value. For a learned numerical interval $[l_k, u_k]$,

$$c_k = \begin{cases} 1, & x_k \in [l_k, u_k], \\ \max\left(0, 1 - \dfrac{\min(|x_k - l_k|, |x_k - u_k|)}{2\max(u_k - l_k, 0.25)}\right), & \text{otherwise.} \end{cases} \tag{4}$$

Previously rejected contexts act as exclusion constraints: a numerical context within 0.08 of a stored mismatch, or an exact match on its Boolean conditions, sets $s_{\omega,i} = 0$.

Candidates with $s_{\omega,i} < 0.75$ or an empty instantiated rewrite are discarded. The remainder are ranked by operator status (*auto-ready* > *validated* > *provisional* > *candidate*) and then by compatibility score. A learned anchor role only disambiguates multiple nodes matched by the same operator. ECHO abstains when different operator families of equal status differ in score by at most 0.05, and each operator can generate at most two proposals per episode.

A newly induced operator is merged with the most similar nondeprecated memory entry when their family similarity reaches 0.80. For family signatures F_1 and F_2, similarity is computed as

$$\mathrm{Sim}(F_1, F_2) = 0.45J_I + 0.15J_R + 0.20J_T + 0.20J_M, \tag{5}$$

where $J(A, B) = |A \cap B|/|A \cup B|$ denotes **Jaccard similarity**, with $J(\emptyset, \emptyset) = 1$. Here, J_I, J_R, J_T, and J_M compare the sets of responder intents, roles, target classes, and coordination motifs, respectively. Operators with different anchor intervention types are incompatible. Two nonempty but disjoint responder-intent sets are also prevented from merging; when both intent sets are empty, disjoint coordination motifs are treated likewise. A successful merge aggregates evidence and context ranges while retaining distinct rewrite variants, with the most frequently observed variant used as the primary executable template. Otherwise, the operator is stored as a new memory entry.

C SUPPLEMENTARY EXPERIMENTAL VISUALIZATION CASES

C.1 SMAC AND OVERCOOKED-AI BENCHMARKS



Figure 6: SMAC tasks: `1c3s5z`, `2m_vs_1z`, `2s_vs_1sc`, `2c_vs_64zg`, `2s3z`, `3s5z`, `3s_vs_3z`, `3s_vs_4z`, and `3s_vs_5z`.

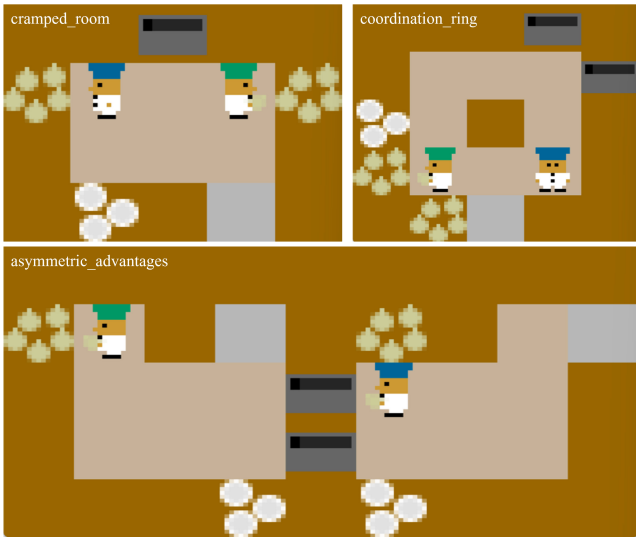

Figure 7: Overcooked-AI tasks: `cramped room`, `asymmetric advantages`, and `coordination ring`.

SMAC. The StarCraft Multi-Agent Challenge (SMAC) requires multiple allied units to cooperatively control their individual units to defeat an opposing army. Under partial observability, agents must coordinate target selection, positioning, and combat actions. An episode succeeds if all enemy units are eliminated within 64 seconds.

Overcooked-AI. Overcooked-AI requires multiple units to collaboratively prepare and deliver dishes by coordinating task allocation, movement, and timing in a shared kitchen. Sequential subtasks, resource contention, and movement conflicts make it suitable for evaluating multi-agent coordination.

C.2 Human Effect on Different Tasks

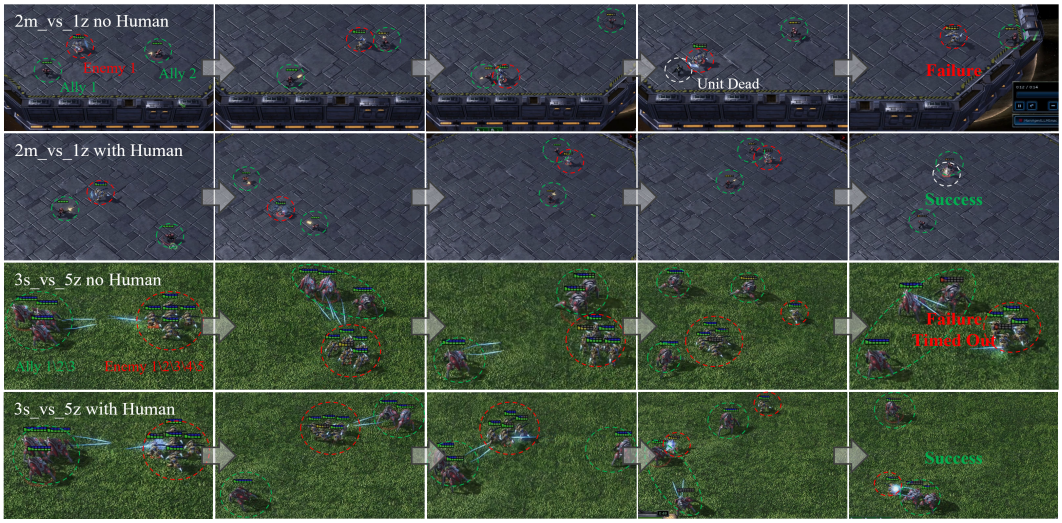

Figure 8: Visualization examples showing that human-guided coordination helps recover successful outcomes in challenging SMAC tasks where autonomous execution fails.



Figure 9: Human guidance enables terrain-aware repositioning on `2c_vs_64zg`, allowing allied units to avoid direct enemy pressure by exploiting cliff and ramp constraints.



Figure 10: Human guidance improves `3s5z` coordination by placing melee units in front to protect key damage dealers, improving survivability and combat efficiency.

Summary C.2. These examples show that human guidance provides task-specific tactical corrections that help recover failures caused by purely autonomous LLM control.

C.3 IMMEDIATE COMPLEMENTARY COORDINATION



Figure 11: Comparison of ECHO-Direct and ECHO-Static, where complementary propagation enables AI-controlled allies to attack while human-controlled units draw enemy attention on the `3s_vs_3z` task.



Figure 12: ECHO-Static propagates human-guided terrain usage to allied units on the `2c_vs_64zg` task, enabling them to avoid enemy attacks more effectively than ECHO-Direct.

Summary C.3. These cases illustrate that immediate human operations are most effective when they are propagated into complementary assignments for the remaining agents. Compared with ECHO-Direct, ECHO-Static turns local interventions into coordinated team responses, improving target focus, unit protection, and terrain-aware positioning.

C.4 SHADOW PROPOSAL AND FEEDBACK EXAMPLES IN SMAC TASKS

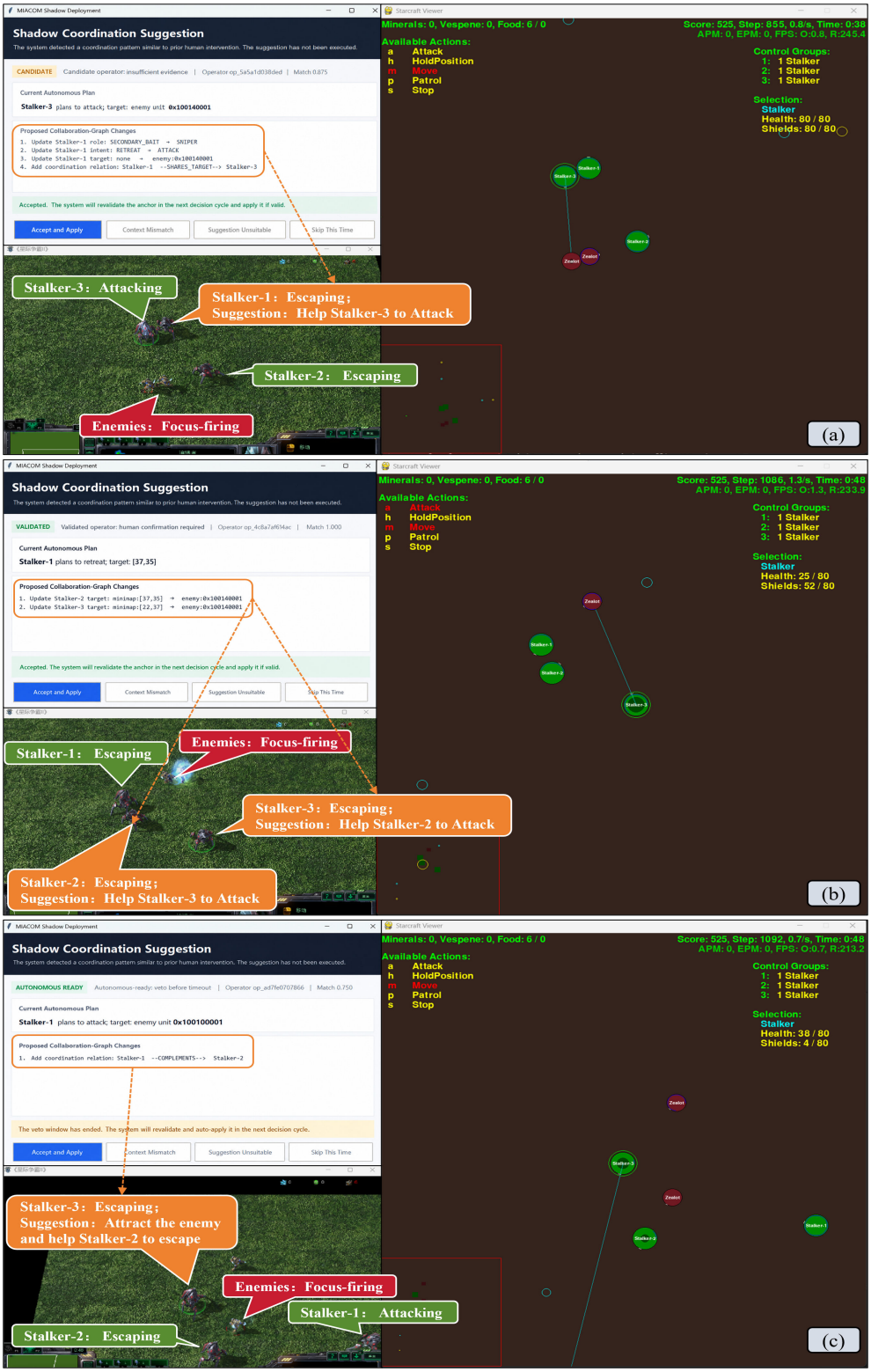

Figure 13: Examples of Shadow coordination suggestions that reuse learned operators (Candidate/-Validated/Auto_Ready) to complement the current autonomous plan under different task states.

C.5 TRANSFERRED OPERATOR REUSE IN SMAC TASKS



Figure 14: Examples of transferred operator reuse, where coordination operators learned on source tasks produce Shadow suggestions for attack, escape, and frontline cover on held-out target tasks.

C.6 Coordination Operators in Overcooked-AI Tasks

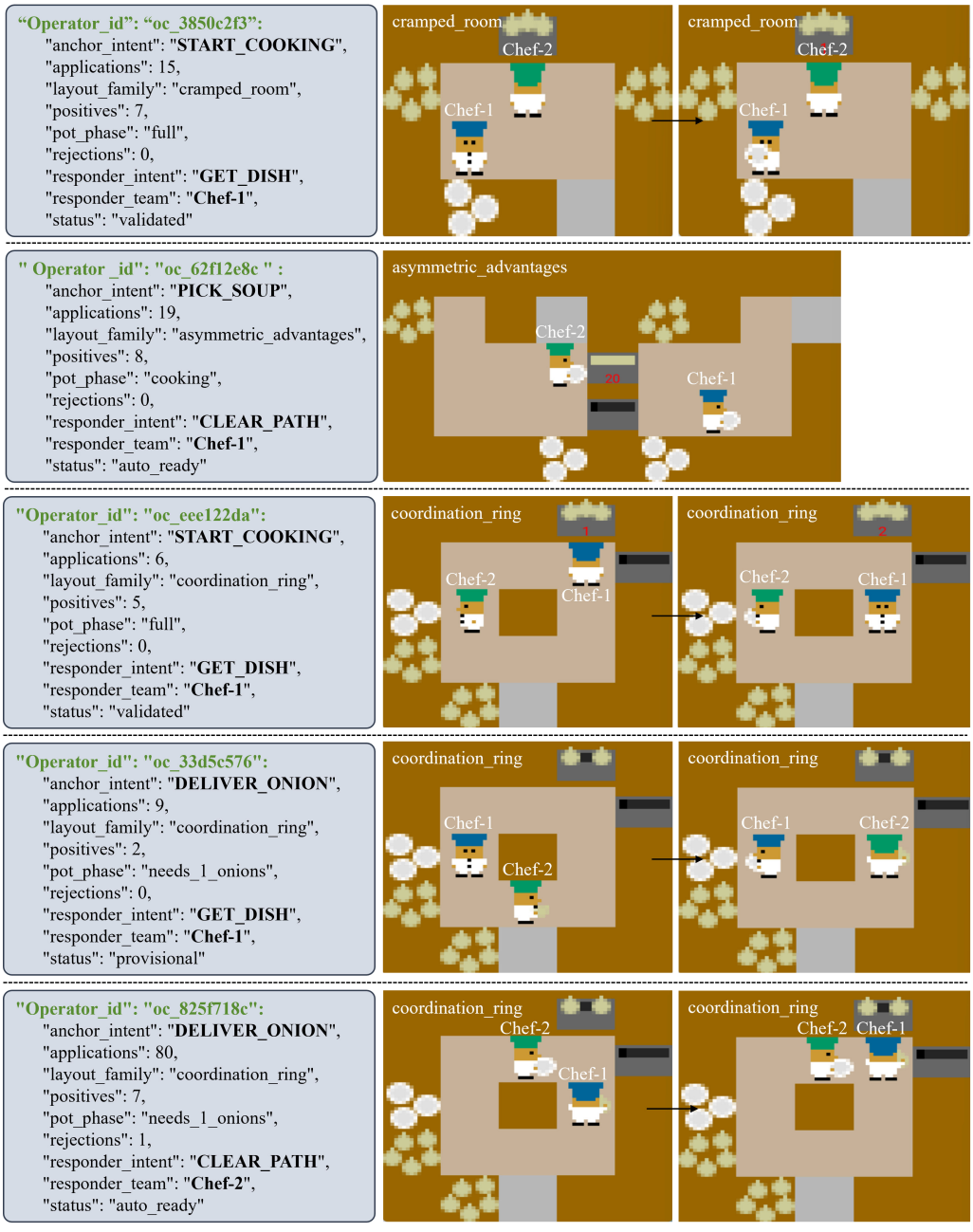

Figure 15: Representative ECHO coordination operators across three Overcooked tasks. The left panels show structured operator records, while the right panels show corresponding coordination states and responder-intent changes.

Summary C.6. Figure 15 illustrates how ECHO stores and reuses context-dependent coordination operators. Each operator links an anchor intent and pot phase to a complementary responder intent, such as `START_COOKING`→`GET_DISH` or `DELIVER_ONION`→`CLEAR_PATH`. The records also report application counts, evidence statistics, and the current evolution status. These examples show that ECHO can capture recurring role-complementary behaviors under different task contexts. The resulting operators provide a compact mechanism for progressively reducing repeated coordination effort.

D HUMAN INTERACTION INTERFACES



Figure 16: Visualized direct human interaction interface of SMAC tasks.

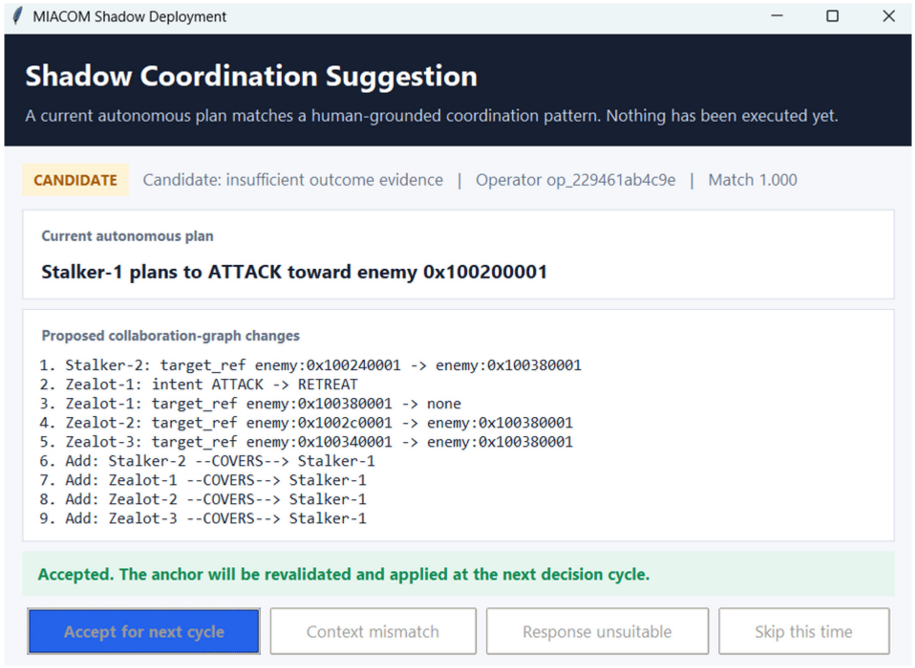

Figure 17: Visualized Shadow human interaction interface.

Figure 16 shows the **direct human-interaction** interface used during real-time control. The interface visualizes ally and enemy positions, unit health, movement and attack dynamics, and a minimap for global spatial awareness. Users can select one or multiple allied units through left-click or drag selection and issue movement or attack commands through right-click operations. These actions are recorded as structured human-operation signals for subsequent intent inference and complementary coordination.

Figure 17 shows the **Shadow interaction** interface used for operator reuse. When a stored coordination operator matches the current decision state, the interface presents the operator status, matching score, current autonomous plan, and the proposed collaboration-graph updates. The human can accept and execute the proposal, mark a context mismatch, reject an invalid proposal, or skip the suggestion. These decisions provide feedback for operator evolution while preventing unmatched or uncertain operators from being applied without supervision.

E LLM ASYNCHRONOUS QUERY CAUSES STATE-ACTION MISMATCH ISSUES

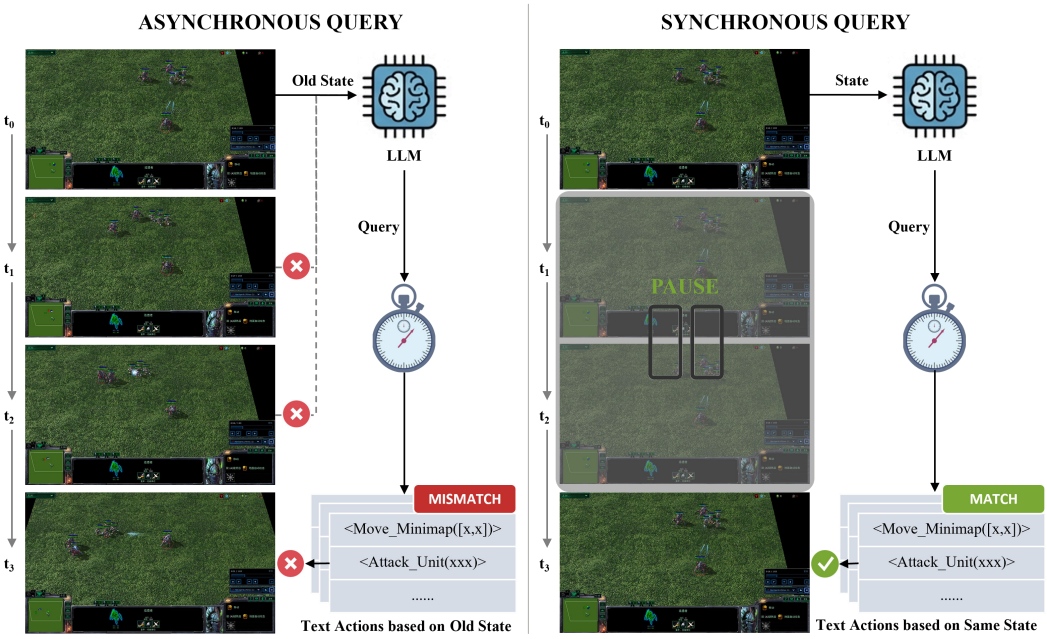

Figure 18: Processes of synchronous query and asynchronous query.

Table 22: Differences between asynchronous and synchronous LLM querying.

Aspect	**Asynchronous Query**	**Synchronous Query**
State evolution	The environment keeps running during LLM inference.	The environment is paused during LLM inference.
Action execution	Actions generated from state s_t may be executed in a later state s_{t+k}.	Actions are executed on the same state used for LLM querying.
Main issue	Observation-action mismatch caused by inference latency.	Reduced mismatch, but with lower runtime efficiency.
Control reliability	Sensitive to fast-changing combat states and delayed commands.	More reliable for fine-grained coordination and target selection.

Figure 18 and Table 22 illustrate the key difference between asynchronous and synchronous LLM querying in SMAC-style multi-agent decision-making. In the asynchronous setting, the environment continues evolving while the LLM is generating actions from the state observed at time t_0. Due to inference latency, the returned text actions are executed at a later time t_3, when the battlefield configuration has already changed substantially. As a result, the generated commands correspond to an outdated state rather than the current one, leading to a clear state-action mismatch. This mismatch may manifest as attacking an invalid target, moving toward obsolete positions, or issuing coordination commands that no longer fit the current combat situation, thereby degrading control reliability. In contrast, the synchronous setting pauses the environment during the LLM query, ensuring that the returned actions are executed on the same state from which they were generated. This design preserves semantic consistency between observation and action, making the control process more stable and interpretable. However, such consistency comes at the cost of lower interaction efficiency, since the environment must wait for the LLM response before proceeding. Overall, the figure highlights a fundamental trade-off in LLM-based control: asynchronous querying improves runtime efficiency

but suffers from observation-action inconsistency, whereas synchronous querying sacrifices speed to guarantee aligned and executable decisions.

To complement the qualitative illustration in Figure 18, we further conduct a diagnostic comparison on representative SMAC maps. The goal of this analysis is not to introduce a new baseline, but to quantify how asynchronous querying can cause state-action mismatch under LLM inference latency.

Table 23: Kill/Death Rate (Success Rate) of Asynchronous and Synchronous Queries.

Method	3s_vs_3z	2s3z	3s5z	2c_vs_64zg	1c3s5z
Asynchronous	0.15(0%)	0.76(20%)	0.47(0%)	0.56(0%)	0.80(30%)
Synchronous	**0.24**(0%)	**0.81**(20%)	**0.49**(0%)	**6.67**(0%)	**0.90**(30%)

To quantitatively analyze the effect of query mode, we compare the asynchronous-query results reported in LLM-PySC2 with our synchronous-query setting. Since the asynchronous-query results are taken from LLM-PySC2 rather than re-run under an identical implementation, this comparison should be interpreted as a diagnostic reference rather than a strictly controlled ablation. Specifically, in our synchronous-query setting, we follow the same experimental configurations as LLM-PySC2, including the evaluated maps, action representation, prompt format, and LLM backbone, with the query mode being the only intended difference. As shown in Table 23, synchronized querying does not necessarily lead to higher success rates. This suggests that simply pausing the environment during LLM inference is insufficient to solve SMAC tasks or explain the main performance improvements of our framework. However, synchronous querying generally yields better Kill/Death Rate, indicating that executing actions on the same state used for LLM reasoning can reduce state-action mismatch and improve combat efficiency. This advantage also comes with an important trade-off: synchronous querying sacrifices runtime efficiency and real-time responsiveness, since the environment must wait for the LLM response before proceeding.

Therefore, we adopt synchronized querying in all reported experiments primarily to ensure semantic consistency between observation and action, rather than as an independent performance-enhancing mechanism.

F Limitations and Future Work

Two limitations define the current scope of ECHO. First, the evaluation is conducted in simulated SMAC and Overcooked-AI environments under synchronized LLM inference, and therefore does not capture the inference delays, evolving states, and execution uncertainties associated with asynchronous real-time deployment. Nevertheless, this controlled setting ensures state–action consistency and enables the coordination mechanisms to be evaluated independently of latency-induced mismatch. Second, transfer and safety are evaluated on related held-out tasks under the defined operator-matching and Shadow-deployment protocol; consequently, the results do not establish cross-domain transfer or safety guarantees beyond the evaluated conditions. Within this scope, the experiments provide reproducible evidence that relational operators support effective coordination reuse while reducing human intervention.

Future work will extend the framework to larger and more heterogeneous teams, asynchronous decision processes, and longer-term operator-memory evolution. Another direction is to develop adaptive matching, operator consolidation, and composition mechanisms for changing team structures and task contexts. We will also investigate richer forms of human feedback and deployment-oriented evaluation to further improve interaction efficiency and reliable coordination.

G OTHER SUPPLEMENTARY MATERIALS

G.1 SYSTEM REPRODUCIBILITY SETTINGS

Table 24: Reproducibility settings for the reported experiments.

Item	Setting
Primary backbone	gpt-5.4, qwen3.8-max
Temperature / samples	0.1 / one response per query
Prompting	Fixed structured prompt shared within each environment
Response format	Structured executable action strings
Maximum output actions	One action per unit or group per query
Timeout / retry limit	15 s / five retries
Query protocol	Synchronized; the environment is paused during each LLM query
SMAC tasks	`1c3s5z`, `2m_vs_1z`, `2s_vs_1sc`, `2c_vs_64zg`, `2s3z`, `3s5z`, `3s_vs_3z`, `3s_vs_4z`, `3s_vs_5z`
Overcooked-AI tasks	`cramped_room`, `asymmetric_advantages`, `coordination_ring`
Immediate coordination	30 matched trials per method–task condition; operator memory disabled
Human study	12 participants; counterbalanced within-subject design; matched scenario seeds across conditions
Continual evolution	Three random seeds, each containing 60 consecutive episodes; six non-overlapping 10-episode windows
Post-evolution evaluation	30 episodes per task using the evolved operator memory
Held-out transfer	30 episodes per source–target pair; source memory frozen and not updated on the target task
Overcooked-AI evaluation	30 trials per method-task combination
Matched controls	Same backbone, prompts, observations, decoding configuration, action interface, and scenario seeds where applicable
Memory initialization	Empty at the beginning of each evolution run and retained across episodes
Transfer protocol	Memory learned on the source task is frozen before target-task evaluation
Reported statistics	Success count/rate; Wilson 95% confidence interval where reported; mean $\pm$ SD for continuous metrics
Human-interaction metric	Direct operations + Shadow confirmations, rejections, context mismatches, and skips
Operator-use metric	Automatic applications
Operating system	Windows 10/11
CPU / GPU	Intel i5-13600KF / NVIDIA RTX 4070 Ti Super (16 GB)
Memory / storage	32 GB RAM / 2 TB SSD
StarCraft II	Version 5.0.15 (96999)
Overcooked-AI	overcooked-ai v1.1.0
Python environment	Detailed configurations are provided in the "Code and configurations"
Code and configurations	`https://anonymous.4open.science/r/-ECHO/`

G.2 Operator Configuration and Ablation Design

Table 25: Thresholds and matching parameters for coordination-operator induction and reuse. Unless otherwise stated, the same settings are used across SMAC and Overcooked-AI.

Parameter	Value	Function
Operator merge threshold	0.80	Merge similar operators
Operator match-score threshold	0.75	Reject incompatible matches
Ambiguity margin	0.05	Abstain on similar scores
Minimum positive outcomes	3	Required for promotion
Minimum distinct positive episodes	2	Required for validation
Maximum negative rate	0.25	Validation safety bound
Deprecation threshold	5 outcomes, $r_\omega > 0.50$	Deprecate unreliable operators
Minimum Shadow confirmations	2	Required for auto-ready status
Auto-ready positive outcomes	3	Required for auto-ready status
Auto-ready negative rate	0.15	Autonomous-reuse safety bound
Response rejection-rate bound	0.20	Bound on valid-context rejections
Auto-ready veto window	15 s	Human veto before auto-application
Context mismatch handling	Exclude from rejection rate	Restrict context without penalty
Operator status order	auto-ready $>$ validated $>$ provisional $>$ candidate	Matching priority

Operators progress from *candidate* to *provisional*, *validated*, and *auto-ready* as positive cross-episode and Shadow evidence accumulates, while unreliable operators are *deprecated*. Reuse filters and ranks operators by match quality and status, abstaining when the match is ambiguous; Table 25 lists the thresholds.

Outcome-evidence instantiation. In SMAC, $\mathcal{V}_t$ requires graph validity, anchor preservation, and selected-unit survival. For $\mathcal{C}_t^\omega$, attack requires target elimination or at least 0.1 damage with a non-negative health trade; retreat requires the nearest-enemy distance to increase by 0.5 or the retreat-point distance to decrease by 0.5; repositioning requires the mean target distance to decrease by 0.5 or end within 0.75; and holding limits displacement to 0.75. Terminal task success also satisfies $\mathcal{C}_t^\omega$. The Overcooked-AI criteria are specified in Appendix B.2.

Table 26: Prompt and adaptation conditions for the evaluated methods. All methods use the same textual observations, legal action interface, and backbone within each environment. "Expert prompt" denotes the shared structured expert-prompt framework, instantiated with task-dependent roles, thresholds, and environment constraints.

Method	Expert prompt	Human operations	Complementary revision	Operator memory	Evidence gating	Auto reuse
LLM-PySC2	✗	✗	✗	✗	✗	✗
HEP (adapted)	✓	✗	✗	✗	✗	✗
ECHO-Auto	✓	✗	✓	✗	✗	✗
ECHO-Direct	✓	✓	✗	✗	✗	✗
ECHO-Static	✓	✓	✓	✗	✗	✗
ECHO-TextRAG	✓	✓	✓	*Text*	✗	✗
ECHO-ConfirmOnly	✓	✓	✓	✓	✓	✗
ECHO-NoSafeguard	✓	✓	✓	✓	✗	✓
ECHO-IDSpecific	✓	✓	✓	✓	✓	✓
ECHO	✓	✓	✓	✓	✓	✓

In Table 26, complementary revision denotes role-aware coordination planning: ECHO-Auto derives it from the expert prior, whereas the interactive variants additionally condition it on a local human operation. Operator memory stores anchor-conditioned responder-side revisions across episodes, while evidence gating covers structured matching, execution feedback, status promotion, rejection handling, and Shadow deployment. The ablations follow a staged logic: LLM-PySC2 versus HEP isolates the expert prompt; HEP versus ECHO-Auto evaluates collaboration-graph planning without human input; and ECHO-Direct versus ECHO-Static isolates propagation of a human operation

to the remaining agents. ECHO-TextRAG evaluates unstructured textual memory, while ECHO-ConfirmOnly evaluates structured operator reuse under mandatory human confirmation. Comparing ECHO-ConfirmOnly with ECHO isolates autonomous reuse. ECHO-NoSafeguard removes evidence gating, and ECHO-IDSpecific replaces relational selectors with concrete identities, testing safety control and transferable abstraction, respectively.

G.3 Prompt Templates

The SMAC expert prompt provides tactical priors while leaving the final action selection state-dependent. Importantly, the expert strategy prompts are not manually designed as task-specific descriptions; instead, they are constructed following the same structured framework of roles, strategy skills, and action constraints, with only task-dependent parameters instantiated accordingly. A shortened representative template is shown below, while the complete task prompts are provided in the anonymous repository.

SMAC Expert Strategy Prompt (Shortened Template)

```
1. Identity
You control a cooperative team on SMAC map {map}.
Each team contains one or more allied units.

2. High-Level Strategy
Follow the task-specific strategy {strategy}.
Use one or more BAIT units to draw enemy pressure while
other units act as SNIPERS or SUPPORT units.
Maintain distance, focus fire, and avoid unnecessary movement.

3. State and Role Assignment
At every step, inspect health, nearest-enemy distance, visible
targets, target health, and previous actions. Assign the closest
threatened team as PRIMARY_BAIT, a SECONDARY_BAIT when appropriate,
and the remaining teams as SNIPERS or SUPPORT units.

4. Priority Rules
For each team, apply this order:
(a) If THREATENED, move to a legal safety waypoint and do not attack.
(b) Else if a valid target is visible, attack immediately and focus
fire on the lowest-health visible enemy when possible.
(c) Else, move to a permitted regroup or search location.
Do not continue a movement loop when an attack is available.

5. Coordination and Human Anchor
Preserve all human-specified anchor units, intents, targets, and
constraints. Assign complementary responses to other teams, such as
support, focus fire, blocking, repositioning, or pressure maintenance.
BAIT units draw pressure while surviving; SNIPER and SUPPORT units
cover the anchor or BAIT and avoid duplicating its escape route.

6. Action Constraints
Output exactly one legal action for each team. Use only valid unit
identifiers, target identifiers, coordinates, and action formats.
Do not invent targets or coordinates.

7. Output Format
Analysis:
  Briefly state the selected role and priority branch for each team.

Actions:
  Team Stalker-1: {action}
  Team Stalker-2: {action}
  Team Stalker-3: {action}
```

The planner receives the current team state and a task-specific expert prior, and returns a schema-validated collaboration graph. Each node specifies a team-level role, intent, subgoal, target, and constraint, while each edge represents an explicit coordination relation. Reasons are restricted to observable facts, and any human-provided anchor is preserved during subsequent graph revision.

SMAC Collaboration-Graph Planner Prompt (Shortened Template)

```
System:
You are the autonomous collaboration-graph planner in ECHO.
Construct a team coordination graph from the current observation and
the expert prior before applying any human intervention.
Use only observable facts in reasons.

Input:
Teams: [Team-1, Team-2, ...]
Game Info: {current observable state}
Expert Strategy: {task-specific strategy prior}
Human Anchor (optional): {protected human operation}

Graph constraints:
Use only legal roles, intents, relations, target references, and
constraints defined by the environment schema. Every team must appear
exactly once as a node. Return valid JSON only; do not add Markdown,
comments, or fields outside the schema.

Output:
{
  "nodes": [
    {
      "team_id": "Team-1",
      "role": "BAIT",
      "intent": "REPOSITION",
      "subgoal": "draw pressure",
      "target_ref": null,
      "constraints": [],
      "reason": "nearest healthy ally"
    }
  ],
  "edges": [
    {
      "source": "Team-2",
      "target": "Team-1",
      "relation": "SUPPORTS",
      "reason": "cover reposition"
    }
  ]
}

Return valid JSON only. Do not add Markdown, comments,
or fields outside the schema.
```

For ECHO, the human anchor and any retrieved operator rewrite are inserted into the Human Anchor and Previous Joint Intent fields. The policy must preserve the anchored chef intent and coordinate the other chef through complementary intents such as obtaining ingredients, fetching dishes, waiting near the pot, clearing a path, or serving soup. The legal candidate set is generated from the current Overcooked state and restricts the output intents for each chef.

Unlike the SMAC controller, the Overcooked policy operates over high-level chef intents rather than unit-level attack or movement actions. The environment adapter subsequently grounds each selected intent into executable Overcooked-AI actions.

Overcooked-AI Prompt (Shortened Template)

```
System:
You are a coordination policy for Overcooked-AI.
Select legal high-level intents for both chefs and return valid JSON
without Markdown.

User:
You control two chefs in Overcooked-AI. Select one high-level intent
for each chef.

Objective:
Deliver {target_deliveries} onion soup(s) before step {horizon}.

Workflow:
Collect three onions, place them in one pot, start cooking if required
    ,
obtain a dish, collect the ready soup, and serve it.

Coordination Rules:
- Do not assign both chefs to a single-capacity interaction when one
    is sufficient.
- Keep narrow passages clear for a chef carrying soup or a dish.
- A chef holding an object must complete or safely yield its object-
    related task.
- Respect the human anchor exactly when one is provided.

Layout:
X/# = counter or wall, O = onion, D = dish, P = pot, S = serving
    counter.
{grid}

State:
{current text observation}

Previous Joint Intent:
{previous joint intent}

Human Anchor:
{chef must execute the anchored intent; coordinate the other chef
    around it}
or None.

Legal Candidates:
{legal candidate intents for Chef-1 and Chef-2}

Return JSON only:
{"Chef-1":"INTENT","Chef-2":"INTENT",
 "reason":"brief coordination rationale"}
```

This example is taken from an actual `3s_vs_5z` execution trace. The graph assigns threatened units to retreat while preserving a long-range attack role for a safe unit. The `SHARES_TARGET` and `COVERS` edges encode focus fire and protection of the retreating bait, respectively. Human-revised graphs additionally record the protected fields in `hard_fields` and mark the modified node fields with `source = human`.

Example Collaboration Graph from SMAC Logs


```
{
  "loop_id": 60,
  "nodes": [
    {
      "team_id": "Stalker-1",
      "role": "SNIPER",
      "intent": "RETREAT",
      "subgoal": "escape melee range",
      "target_ref": null,
      "constraints": ["maintain distance > 4"],
      "reason": "Nearest enemy distance is 2; the unit is threatened
    and must retreat.",
      "source": "autonomous"
    },
    {
      "team_id": "Stalker-2",
      "role": "PRIMARY_BAIT",
      "intent": "RETREAT",
      "subgoal": "escape melee range",
      "target_ref": null,
      "constraints": ["maintain distance > 4"],
      "reason": "Nearest enemy distance is 3; the unit is threatened
    and requires immediate escape.",
      "source": "autonomous"
    },
    {
      "team_id": "Stalker-3",
      "role": "SNIPER",
      "intent": "ATTACK",
      "subgoal": "focus fire weak enemy",
      "target_ref": "enemy:0x100100001",
      "constraints": ["maintain range >= 6"],
      "reason": "The unit is in the FIRE_ZONE and targets the lowest-
    health visible enemy.",
      "source": "autonomous"
    }
  ],
  "edges": [
    {
      "source": "Stalker-1",
      "target": "Stalker-3",
      "relation": "SHARES_TARGET",
      "reason": "The snipers focus fire on the same weak enemy.",
      "provenance": "autonomous"
    },
    {
      "source": "Stalker-1",
      "target": "Stalker-2",
      "relation": "COVERS",
      "reason": "The sniper covers the retreating bait unit.",
      "provenance": "autonomous"
    },
    {
      "source": "Stalker-3",
      "target": "Stalker-2",
```

```
      "relation": "COVERS",
      "reason": "The sniper attacks the shared threat while the bait
    retreats.",
      "provenance": "autonomous"
    }
  ],
  "hard_fields": [],
  "provenance": {
    "planner_format": "json",
    "source": "autonomous",
    "validation": {
      "valid": true,
      "human_fidelity": true,
      "fallback_used": false
    }
  }
}
```


Human-Anchored Graph Revision


```
{
  "human_anchor": {
    "team_ids": ["Stalker-1", "Stalker-2", "Stalker-3"],
    "intent": "ATTACK",
    "target_ref": "enemy:0x1001c0001"
  },
  "hard_fields": [
    ["Stalker-1", "intent"],
    ["Stalker-1", "target_ref"],
    ["Stalker-2", "intent"],
    ["Stalker-2", "target_ref"],
    ["Stalker-3", "intent"],
    ["Stalker-3", "target_ref"]
  ],
  "revision_rule": "Preserve the human anchor and assign complementary
roles and relations to the remaining graph fields."
}
```