TRACE-EVO: Contribution-Driven Role Reassignment and Knowledge Transfer in Long-Horizon Multi-Agent Systems

Anonymous submission

Abstract

Large language model agents are increasingly organized as role-specialized teams with persistent memories and reusable skills. However, most multi-agent systems either fix role assignments before work begins or adapt only transiently within a single task, leaving persistent agent–role mismatch unresolved as task demands and role-local experience evolve. We study long-horizon heterogeneous multi-agent systems under fixed model identities, where organizational improvement must come from changing responsibilities rather than upgrading agents. We introduce TRACEBench, a stateful enterprise-style benchmark with explicit roles, mutable business state, and a growth-then-freeze protocol that evaluates whether an organization formed during prior work transfers to held-out tasks. We further propose TRACE-EVO, a contribution-driven framework that updates agent and role profiles from verified outcomes, searches bounded role reassignments, validates candidate changes through real-work probation, and transfers role-scoped memories and skills through governed handoff with rollback and re-certification. Across 12 TRACEBench cases, three initialization profiles, and four heterogeneous model pairs, TRACE-EVO improves frozen held-out performance over Static-Growth by 0.26–1.75 points and reduces catastrophic failures across all model pairs, while remaining competitive with workflow-memory and skill-transfer baselines. These results suggest that persistent organizational state is a key evaluation and design object for long-horizon multi-agent systems.

Introduction

Large language model (LLM) agents are increasingly organized into teams with specialized roles, such as planning, execution, review, and coordination (Wu et al. 2024; Li et al. 2023; Hong et al. 2024; Chen et al. 2024; Qian et al. 2024). A central design decision in such systems is how to assign heterogeneous agents to these roles. Before the target work is performed, existing systems typically rely on model identity, public benchmark performance, manually specified expertise, or historical results from related tasks. These signals provide useful priors about general capability, but they do not directly reveal how an agent will perform in a particular role under the target task, tools, team dependencies, and operational constraints. Meanwhile, the practical demand of a role cannot be inferred from its title alone, because different roles become critical bottlenecks under different tasks and stages of execution. The initial agent–role assignment is therefore constructed from incomplete and indirect evidence.

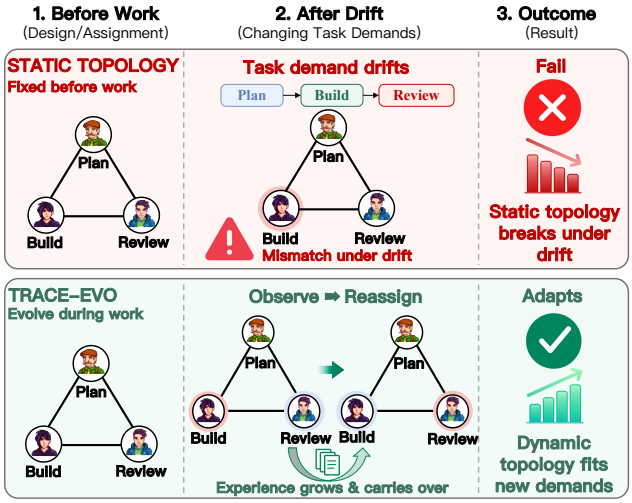

Figure 1: Static role assignments become mismatched as task demands and observed agent capabilities evolve. TRACE-EVO revises agent–role assignments from verified work evidence and validates candidate changes before commitment.

This limitation becomes more serious in long-horizon settings. A long-running team does not repeatedly solve an unchanged task: task requirements, tool conditions, resource constraints, and downstream dependencies evolve, shifting both the demand and importance of different roles. At the same time, agents accumulate different role-specific experience, memories, and reusable skills through the work they perform. Agent–role fit is therefore not fixed after initialization. An assignment that is reasonable at the beginning becomes outdated as the environment and the team evolve. Nevertheless, many existing multi-agent systems treat role assignment as a one-time pre-execution decision and keep the resulting topology fixed throughout subsequent work,

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

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.

preventing newly observed evidence from influencing persistent responsibility allocation.

These observations raise a fundamental challenge: how can a long-running team determine when its current organization has become unsuitable and revise its persistent role assignment from actual work evidence? Unlike per-task routing or workflow search, which selects a temporary collaboration pattern for the current input, this setting requires updating an organization whose assignments, accumulated experience, and downstream effects persist across subsequent tasks. The goal is therefore not simply to find a better topology once, but to support reliable organizational evolution as agent–role fit changes over time.

To address this challenge, we propose TRACE-EVO, which treats role assignment as a persistent organizational state revised through verified work evidence. TRACE-EVO updates agent profiles and role-demand profiles from observed outcomes, constructs capability- and experience-aware matching scores, and searches for bounded candidate reassignments. Rather than committing a candidate immediately, TRACE-EVO evaluates it through real-work probation. Changes supported by subsequent outcomes are committed, while unsupported changes are rolled back. During successful transitions, validated role-relevant experience is retained to preserve work continuity, while each agent's underlying model identity remains fixed.

Evaluating such evolution requires more than measuring success on isolated tasks. Existing agent benchmarks commonly evaluate independent episodes or permit continued adaptation during testing, making it difficult to determine whether a team has formed a transferable organization. We therefore introduce TRACEBench, a stateful benchmark with heterogeneous agents, explicit role responsibilities, and fixed model identities. TRACEBench adopts a growth-then-freeze protocol: teams accumulate experience and revise role assignments during growth, but roles, knowledge, and organizational structure are frozen before held-out evaluation. This protocol isolates organizational transfer from continued test-time adaptation and measures whether the organization formed during prior work generalizes to new tasks.

Across 12 TRACEBench cases, three initialization conditions, and four heterogeneous model pairs, TRACE-EVO consistently improves frozen held-out performance over Static-Growth and reduces catastrophic failures. Further analyses show that it remains stable under well-matched initializations while adapting effectively to partial and global role mismatch.

Our contributions are threefold:

- We formulate long-horizon multi-agent collaboration as a persistent agent–role assignment problem in which role fit is initially uncertain and changes throughout repeated work.
- We introduce TRACEBench, a stateful benchmark with heterogeneous agents, controlled initial assignments, fixed model identities, and a growth-then-freeze protocol for evaluating organizational transfer.
- We propose TRACE-EVO, an evidence-driven framework that revises persistent role assignments through capability-aware matching, real-work probation, verification, rollback, and validated experience retention.

Related Work

Role, Memory, and Organizational Control LLM-based multi-agent systems commonly use specialized roles and structured collaboration patterns (Wu et al. 2024; Li et al. 2023; Hong et al. 2024; Qian et al. 2024; Chen et al. 2024). Related work further optimizes agent graphs, workflows, or communication structures at test time or search time (Zhuge et al. 2024; Zhang et al. 2025; Wang et al. 2025; Liu et al. 2024b), while persistent-memory and skill-based agents reuse experience across interactions (Yao et al. 2023; Shinn et al. 2023; Park et al. 2023; Wang et al. 2024; Packer et al. 2024; Zhong et al. 2024). TRACE-EVO differs by treating role assignment itself as persistent organizational state: model identity is fixed, role responsibility changes through governed reassignment, and role-local memory or skills must be transferred through a verified handoff rather than simply shared.

Another relevant line of work studies credit assignment in cooperative multi-agent settings, where the aim is to estimate how individual agents or actions contribute to a shared return. Representative approaches include Shapley values, difference rewards, counterfactual baselines, value decomposition, and monotonic value factorization (Shapley 1953; Wolpert and Tumer 2002; Foerster et al. 2018; Sunehag et al. 2018; Rashid et al. 2018). TRACE-EVO uses contribution evidence for a different decision: not to decompose a single reward, but to decide whether persistent role responsibilities should be revised under fixed agent identities. Accordingly, our method treats credit signals as auditable inputs to agent and role profiles, and commits a role change only after the proposed organization is validated through subsequent work.

Long-Horizon Agent Benchmarks Recent benchmarks evaluate LLM agents on tool use, web navigation, software engineering, operating-system control, machine-learning experimentation, and general assistant tasks (Qin et al. 2024; Liu et al. 2024a; Zhou et al. 2024; Jimenez et al. 2024; Yang et al. 2024; Xie et al. 2024; Mialon et al. 2024; Huang et al. 2024). WorkArena, τ-bench, and TheAgentCompany move closer to realistic workplace or tool-user interaction settings (Drouin et al. 2024; Yao et al. 2025; Xu et al. 2025). TRACEBench complements these benchmarks by freezing a stateful team after growth, so held-out performance reflects transferred organizational capability rather than continued online adaptation.

TRACEBench

TRACEBench is designed to evaluate organizational evolution rather than one-shot task solving. Its construction is inspired by TheAgentCompany (TAC) (Xu et al. 2025), which builds hand-curated enterprise workflows in a controlled workplace environment. Following this direction, we use GPT-5.5 to draft the high-level enterprise scenario framework, including task-family coverage, company state, role structure, tool semantics, and anticipated failure modes; human authors then manually instantiate the task details, ev-

idence objects, state transitions, role responsibilities, and evaluation checks. A benchmark instance is therefore a simulated enterprise case with mutable business state, role-specific responsibilities, persistent memories, reusable skills, and controlled tools. A team first completes a sequence of work-phase tasks in which methods accumulate permitted knowledge and revise role assignments under the permitted protocol. The resulting organization is then frozen and evaluated on held-out business tasks. This separation lets us test whether a team has built transferable organizational capability, instead of merely continuing to adapt during evaluation.

Stateful Enterprise Cases TRACEBench contains 12 enterprise cases grouped into six task families, with two cases per family. The cases cover the kinds of work where long-horizon agent teams are expected to accumulate procedural knowledge: release artifacts, incident response, data reconciliation, configuration operations, evidence synthesis, and resource planning. Each case uses the same five-role team structure but different company entities, constraints, tool semantics, and failure modes.

TRACEBench tasks are not pure text role play. Each task is executed through constrained procedures that inspect evidence, modify simulated business state, update documents, and manage memory or skill objects. The environment then applies deterministic checks to the resulting state. These checks cover code or configuration changes, data and resource constraints, evidence-backed documents, memory versioning, skill certification, role ownership, reviewer state, and authorization state. Each checked requirement is also assigned to an owner role through a RoleCard, enabling role-level analysis in addition to the 0–10 hard task score.

Agents and Roles Each case uses five agents and five non-substitutable roles: Leader, Analyst, Engineer, Liaison, and Reviewer. The roles are intentionally not ordered by a fixed global importance rule. Depending on the case, engineering execution, evidence review, stakeholder coordination, or risk validation becomes the bottleneck. This design avoids reducing organizational evolution to the special case of always placing the strongest model in the leader role.

The team is heterogeneous: two agents are bound to a stronger model and three agents are bound to a weaker model. Model identity is attached to the agent, not to the role. The protocol permits role reassignment and validated knowledge transfer, but prohibits upgrading a weak agent into a strong model or using a stronger model to repair that agent's failed output. This constraint is central to TRACEBench because the benchmark aims to measure organizational improvement under fixed personnel capability.

Work Phase and Frozen Evaluation Each case contains 24 work-phase tasks followed by 12 frozen evaluation tasks. The work phase is divided into four six-task segments. Tasks 1–6 establish the operational base through evidence gathering, first execution, initial documentation, and early memory or skill creation. Tasks 7–12 introduce the first migration pressure, such as changed rules, changed tool semantics, stale experience, or new business units. Tasks 13–18 stress responsibility transfer through policy changes, conflicting metrics, skill review, absence of a key role holder, and handoff. Tasks 19–24 test stabilization under old-knowledge interference, resource constraints, new tool versions, independent audit, and combined delivery.

The frozen phase uses the same company state and historical artifacts but introduces new business combinations rather than exam-style questions. Examples include a new unit migration combined with stale process cues, vendor failure under reduced capacity, conflicting evidence with partial rollback, leader absence, and tools that report success while the underlying state check still fails. During frozen evaluation, the team is permitted to read valid knowledge produced during the work phase, but is prohibited from creating or editing memory, creating or certifying skills, changing roles, changing the leader, modifying reviewer assignments, or altering communication and permission structure.

We describe the initialization sampling and reporting metrics in the experiment setup.

TRACE-EVO

Figure 2 gives an overview of TRACE-EVO, an organizational evolution algorithm for long-horizon multi-agent teams. The framework treats each agent as a fixed employee-like identity with an immutable underlying model, while its organizational position, or role, changes through accepted reassignment decisions. During work, an agent accumulates role-local experience, acquires reusable skills, and maintains documents or memories for the position it currently holds. TRACE-EVO decides when the current organization no longer matches the work demand and how to restructure role assignments without changing any agent's intrinsic capability. The method has three components: fixed identity and organizational state, contribution-driven role reassignment, and transactional knowledge handoff.

Fixed-Identity Organizational State

Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be the agents and $\mathcal{R} = \{r_1, \ldots, r_n\}$ be the organizational roles. Each agent a represents a persistent employee-like identity with a fixed model identity $m(a)$ and a fixed capability tier. At time t, the organization is a role assignment $\pi_t : \mathcal{R} \to \mathcal{A}$, analogous to assigning employees to positions. TRACE-EVO modifies π_t only through accepted reassignment decisions, and it never modifies $m(a)$:

$$m_t(a) = m_0(a), \quad \forall a \in \mathcal{A}, \forall t. \tag{1}$$

This constraint prevents organizational gains from being conflated with hidden model upgrades or implicit changes to personnel capability.

The remaining organization state is represented as structured records rather than free-form summaries. An agent profile contains fixed fields and learned counters:

$$\begin{aligned} P_t(a) = \{&\tau(a), z(a), n_t(a,r), \bar{s}_t(a,r), \\ &u_t(a,r), \ell_t(a), c_t(a), K_t(a)\}_{r \in \mathcal{R}}. \end{aligned} \tag{2}$$

Here $\tau(a)$ is the fixed capability tier, $z(a)$ is a fixed employee profile used for role-fit matching, $n_t(a,r)$ is the number of direct public observations for (a,r), $\bar{s}_t(a,r)$ is the corresponding public role score, $u_t(a,r)$ is uncertainty, $\ell_t(a)$ is

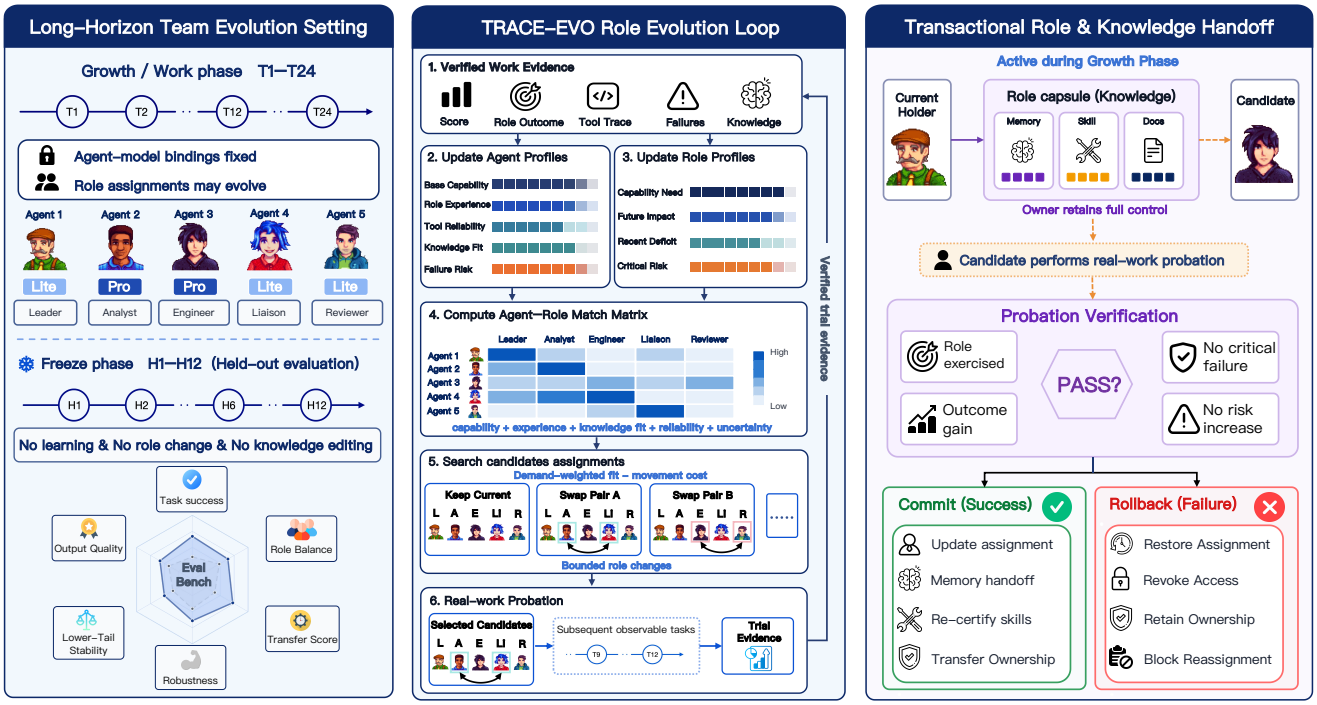

Figure 2: Overview of TRACE-EVO. The framework updates persistent role assignments from verified work evidence, tests candidate changes through probation, and transfers role-specific knowledge only after validation.

tool or action reliability, $c_t(a)$ is catastrophic-failure rate, and $K_t(a)$ indexes owned knowledge assets. Long natural-language memories and documents are referenced by asset identifiers rather than inserted into the matching score.

Initialization is deterministic. The run configuration binds two agents to the stronger model tier and three to the weaker tier. Before any work is observed, $n_0(a, r) = 0$ for every pair. Capability priors use the same 0–10 scale as task scores: stronger-tier agents start at 7.0 and weaker-tier agents at 5.5. Public role reliability starts neutral, catastrophic rate starts at 0, and uncertainty starts high because no agent-role pair has direct evidence.

Each role also has a structured profile

$$Q_t(r) = \{p(r), D_0(r), I_t(r), B_t(r), C_t(r)\}. \tag{3}$$

The fixed part contains the required role profile $p(r)$ and a declared capability demand $D_0(r)$ from the case specification. The submitted controller clips $D_0(r)$ to $[0.10, 1.50]$ and defaults to 1.0 when absent. The dynamic part contains causal importance $I_t(r)$, recent bottleneck indicators $B_t(r)$, and critical-check evidence $C_t(r)$, all updated only from completed public work traces. Knowledge assets are similarly typed records with owner, target role scope, world version, parent link, validity status, and certification state. When a role changes, TRACE-EVO transfers governed access to these records while keeping fixed agent identity and role assignment as separate objects.

Contribution-Driven Role Reassignment

TRACE-EVO updates staffing decisions from verified work evidence. After each task, the controller receives only public trajectory feedback: the task score band, the role and agent for each stage, visible action outcomes, blocked-by-upstream status, knowledge operations, and public failure summaries. It does not read hidden evaluator answers or frozen-task labels.

For an executed stage, TRACE-EVO forms a local public observation. If the stage is not blocked by an upstream role, its observation score is

$$o = 0.65 \cdot \text{ActionSuccess} + 0.35 \cdot \text{TaskBand}, \tag{4}$$

where TaskBand maps pass, partial, and fail to 1.0, 0.45, and 0.0. This value updates $n_t(a, r)$ and $\bar{s}_t(a, r)$ for the agent that actually held the role. Evidence is sparse by a fixed count rule: zero direct unblocked observations trigger backoff to related role-profile evidence or the fixed capability prior; fewer than two direct observations are low confidence. The uncertainty term uses the smoothed count penalty $u_t(a, r) = 0.80/\sqrt{n_t(a, r) + 1}$.

The central decision object is an agent-role matching matrix. In the submitted controller, the score for assigning agent a to role r is instantiated as

$$M_t(a, r) = D_0(r) I_t(r) \big(0.65\,\text{cap}(a) + 0.35\,\text{fit}(a, r) + 0.04\,\gamma_t(a, r) + 0.02\,\rho_t(a, r)\big). \tag{5}$$

Here $\text{cap}(a)$ is fixed model capacity (1.0 for a stronger-tier agent and 0.58 for a weaker-tier agent), $\text{fit}(a, r)$ is declared

employee-to-role fit, $\gamma_t(a, r)$ is confidence-weighted role evidence, and $\rho_t(a, r)$ is bounded public reliability. All weights are fixed before evaluation and shared across cases and model families. The large explicit capacity term prevents early experience in an easy role from permanently blocking a stronger fixed-model agent from taking over a high-demand role.

Role demand is task-dependent. TRACE-EVO starts from the declared demand $D_0(r)$ and multiplies it by a causal importance term computed from recent public failures:

$$I_t(r) = 1 + 0.35\, F_t(r) + 0.65\, P_t(r), \tag{6}$$

where $F_t(r)$ is the recent direct failure rate of role r, and $P_t(r)$ is the fraction of downstream stages visibly blocked after a direct failure by r. This design makes the important role data-dependent: the strongest agent is not assumed to belong to the Leader role unless the public trajectory makes leadership the current bottleneck.

At organization-review checkpoints, TRACE-EVO enumerates candidate assignments in the five-agent permutation space and selects at most one reversible trial. The objective is

$$\begin{aligned} V_t(\pi) = {} & \textstyle\sum_{r \in \mathcal{R}} M_t(\pi(r), r) \\ & -\lambda c_t(\pi) + \rho b_t(\pi). \end{aligned} \tag{7}$$

The submitted controller filters this objective through hard gates before a trial starts. There must be an observed problem role; the proposed reassignment must repair the identified bottleneck or move the team closer to a predefined capability reference; ordinary trials are bounded to two- or three-role cycles; directly validated high-demand roles are protected; and failed candidate signatures are blacklisted for the rest of the run. The gain must clear a fixed threshold (0.025 in the controller), and the changed roles must have upcoming ordinary work that observes the trial.

Role evolution is not committed immediately. A selected candidate enters real-work probation under ordinary subsequent tasks. For each changed role, TRACE-EVO collects two direct unblocked observations under the proposed holder. The validation score is the mean of these observations, weighted by $D_0(r)I_t(r)$. A trial commits only if no high-impact changed role regresses, and either the weighted post-trial utility is not materially worse than before (within 0.08) or a high-impact role improves by at least 0.10. A direct catastrophic cascade triggers early rollback. On rollback, the previous assignment is restored, delegated access is revoked, and the failed candidate is added to the run-local exclusion set.

Transactional Knowledge Handoff

Role changes are unsafe when the previous holder owns critical memories, skills, or documents accumulated while occupying that position, even if the incoming holder has higher base capability. TRACE-EVO therefore treats knowledge transfer as a transaction rather than unrestricted sharing. Before probation, the system constructs a Role Capsule for each changed role. The submitted controller selects at most two current-world role memories and one operational role skill from the incumbent. Each selected item must be active, role-scoped, and compatible with the current world version; unrelated private memories and expired knowledge are excluded.

During probation, the candidate receives only transaction-scoped access to capsule contents. Memory copies keep provenance through parent identifiers. Skill copies are marked as pending for the receiver: they guide the trial, but they do not count as certified competence until a later valid task re-certifies them. If the candidate fails, TRACE-EVO rolls back the role assignment, revokes the delegated or pending capsule state, and records the failed assignment.

If probation succeeds, TRACE-EVO commits the role change and then updates knowledge ownership. Memories are copied with parent links and version metadata. Skills are not blindly copied as executable competence; they must be re-certified using evidence from the probation window or a later valid task. This separation preserves useful organizational knowledge while limiting stale or unverified transfer.

Experiments

Experiment Setup

We evaluate each method on the full TRACEBench suite: 12 enterprise cases with three fixed initialization profiles, yielding 36 paired units per method and model pair. Each run contains 24 work-phase tasks followed by 12 frozen held-out tasks. The frozen phase is the primary evaluation phase because roles, knowledge, and communication structure are locked before held-out tasks begin.

Initialization Sampling For each case, we define three fixed initial role assignments through a predefined stratified sampling protocol. The profiles cover a near-fit organization, a partial-mismatch organization, and a global-mismatch organization, testing stability, local repair, and severe-mismatch recovery respectively. All methods are paired on exactly the same case and initialization profile; the calibration sweep and sampling details are provided in the appendix.

Baselines We compare TRACE-EVO with one control baseline and two strong baselines adapted from prior work. Static-Growth keeps the initial role assignment fixed while allowing agents to accumulate knowledge in their assigned roles. Shared-Workflow-Memory adapts persistent memory and reflection mechanisms (Shinn et al. 2023; Park et al. 2023; Packer et al. 2024; Zhong et al. 2024) by extracting successful workflows into team-visible memory without role changes. Skill-Library-Transfer adapts reusable skill-library agents such as Voyager (Wang et al. 2024) by extracting validated operations into transferable skills without role evolution. All methods share the same initialization, task order, retry policy, and fixed agent-model identities.

The completed main runs cover four heterogeneous model pairs: Dola Seed 2.0 Pro/Lite, DeepSeek V4 Pro/Flash, MiniMax M3/M2.7, and GPT-5.5/Dola Seed 2.0 Lite. In each pair, two agents are bound to the stronger model and three agents to the weaker model.

Metrics The primary metric is *HeldoutMean*, the mean hard score over the 12 frozen business tasks. The main table also reports Work and Overall scores, BalancedTrans-

Method	Score				vs. Static			Risk/Cost	
	Work	Held.	Overall	BT	ΔH	Win	N.W.	Cat.	Org.
Dola Seed 2.0 Pro/Lite									
Static-Growth	3.31	3.24	3.28	2.31	–	–	–	0.56	0.00
Shared-Memory	3.41	3.19	3.33	2.28	-0.05	11/36	19/36	0.58	0.00
Skill-Transfer	4.10	4.49	4.23	3.35	+1.25	29/36	33/36	0.22	0.00
TRACE-EVO	**5.26**	**4.70**	**5.08**	**3.56**	**+1.47**	**32/36**	**34/36**	**0.12**	2.33
DeepSeek Pro/Flash									
Static-Growth	2.57	2.57	2.57	1.74	–	–	–	0.72	0.00
Shared-Memory	2.55	2.52	2.54	1.67	-0.05	8/36	24/36	0.75	0.00
Skill-Transfer	2.58	2.45	2.54	1.64	-0.12	10/36	19/36	0.68	0.00
TRACE-EVO	**4.60**	**3.44**	**4.21**	**2.39**	**+0.87**	**25/36**	**27/36**	**0.30**	4.53
MiniMax M3/M2.7									
Static-Growth	2.75	2.63	2.71	1.79	–	–	–	0.70	0.00
Shared-Memory	2.80	2.69	2.76	1.80	+0.06	17/36	23/36	0.67	0.00
Skill-Transfer	2.62	2.50	2.58	1.71	-0.13	14/36	18/36	0.67	0.00
TRACE-EVO	**3.06**	**2.90**	**3.00**	**2.00**	**+0.26**	**21/36**	**22/36**	**0.56**	6.36
GPT-5.5/Dola Seed 2.0 Lite									
Static-Growth	4.98	5.07	5.01	4.29	–	–	–	0.41	0.00
Shared-Memory	5.02	5.13	5.06	4.32	+0.06	14/36	24/36	0.40	0.00
Skill-Transfer	5.70	5.64	5.68	4.66	+0.57	25/36	27/36	0.16	0.00
TRACE-EVO	**6.48**	**6.83**	**6.60**	**5.99**	**+1.75**	**32/36**	**34/36**	**0.13**	1.94

Table 1: Main results on TRACEBench. Each row aggregates 36 paired units. Held. is the frozen held-out mean and is the primary metric. BT denotes BalancedTransfer. ΔH, Win, and N.W. compare each method with Static-Growth under the same model pair; N.W. counts paired units where the method is no worse than Static-Growth. Cat. denotes catastrophic failure rate, and Org. denotes organization cost.

Variant	Held.	Over.	BT	Cat.	Org.
Static-Growth	3.24	3.28	2.31	0.556	0.00
Role Evolution Only	4.20	4.30	3.07	0.218	5.83
w/o Rollback	4.59	4.84	3.48	0.120	1.25
w/o Skill Re-cert.	4.66	5.00	3.54	0.118	2.44
TRACE-EVO	**4.70**	**5.08**	**3.56**	**0.116**	2.33

Table 2: Component ablation on TRACEBench with Dola Seed 2.0 Pro/Lite. Held. is the primary frozen held-out score; Over. is the overall score. Lower Cat. and Org. are better.

fer (BT), the held-out difference from Static-Growth (ΔH), paired Win and non-worse counts (N.W.), catastrophic failure rate (Cat.), and organization cost (Org.). A catastrophic failure is a task score no higher than 2 out of 10. RoleCards provide role-level statistics, including weakest-role performance and role dispersion, so a high-performing role cannot hide another role's failure. Full metric formulas, process-count definitions, and diagnostic fields are given in the appendix.

Main Results

Table 1 shows that TRACE-EVO obtains the highest held-out score under all four completed model pairs and consistently improves over Static-Growth. Relative to Static-Growth, held-out performance improves by 1.47 points for Dola Seed 2.0, 0.87 points for DeepSeek, 0.26 points for MiniMax, and 1.75 points for GPT-5.5/Dola Seed 2.0 Lite. In paired units, TRACE-EVO is no worse than Static-Growth in 34/36, 27/36, 22/36, and 34/36 units for the four model pairs, respectively. TRACE-EVO also reduces catastrophic failures in all four settings, with the largest reductions under Dola Seed 2.0, DeepSeek, and GPT-5.5/Dola Seed 2.0 Lite.

The comparison with knowledge-transfer baselines is more nuanced. Under Dola Seed 2.0, Skill-Library-Transfer is already strong, and TRACE-EVO improves held-out performance by only 0.21 points over it. Under DeepSeek and GPT-5.5/Dola Seed 2.0 Lite, TRACE-EVO is higher than all three baselines by larger margins. Under MiniMax, the average improvement is modest, but TRACE-EVO still achieves the best held-out score and lowest catastrophic failure rate. These results show that role evolution is most useful when static organizations fail to place stronger fixed-model agents in roles that become capability-intensive, while knowledge-transfer mechanisms explain part of the gain.

Component Ablation

We ablate the main components under the Dola Seed 2.0 Pro/Lite model pair, using the same 12 cases and three initialization profiles. Role Evolution Only disables role-memory

Variant	Held.	BT	Cat.	Org.	Trial
Full TRACE-EVO	**4.70**	**3.56**	0.12	2.33	0.81
Experience-First	4.58	3.43	**0.09**	**0.67**	0.42
Capability-Only	4.53	3.41	0.11	1.50	1.00
No-CapacityAlign	4.61	3.48	0.13	1.50	1.03

Table 3: Matching signal ablation on TRACEBench with Dola Seed 2.0 Pro/Lite. Values are means over 36 paired units per variant. Lower Cat., Org., and Trial are better when scores are comparable.

Init.	N	ΔH	ΔBT	ΔCat.	Org.
Fit	48	+1.06	+0.83	-0.39	0.17
Partial	48	+1.05	+0.92	-0.33	2.69
Global	48	+1.16	+1.10	-0.24	8.52

Table 4: Initialization-stratified analysis across four model pairs. Δ values compare TRACE-EVO with Static-Growth on matched cells. Org. is the mean organization cost.

and skill handoff, w/o Rollback removes recovery after a failed trial, and w/o Skill Re-cert. reuses transferred skills without re-certification.

Table 2 shows that reassignment alone is weaker than TRACE-EVO, indicating that role-local knowledge must move with the role. The rollback and skill re-certification ablations are closer to the full method, showing that these governance steps mainly improve reliability, while the largest component effect comes from coupling reassignment with governed knowledge handoff.

Analysis

Matching Signal Ablation We next isolate the matching signals used by TRACE-EVO. All variants keep the same role-evolution, handoff, and validation protocol, but change how the agent-role matching matrix is computed. Experience-First increases the coefficients on incumbent role history and observed same-role evidence. Capability-Only uses a capability-dominant score centered on each agent's fixed model tier. No-CapacityAlign removes the explicit role-demand and agent-capacity alignment term. Each variant is evaluated with Dola Seed 2.0 Pro/Lite on the same 12 cases and three initialization profiles.

Table 3 shows that no single signal matches the full controller. Capability-only matching and removing capacity alignment both reduce Held. and BT, while Experience-First lowers organization cost at the expense of transfer performance. The submitted matching function therefore combines fixed capacity, role-specific evidence, role demand, and process reliability.

Initialization-Stratified Behavior TRACEBench includes three fixed initialization profiles per case. We aggregate paired TRACE-EVO versus Static-Growth cells across the four completed model pairs to test whether the method depends on a particular initial assignment. Table 4 shows gains in all three strata. The global-fit setting improves with little reorganization, while global mismatch produces the largest gain and highest organization cost.

Family	N	ΔH	Win	ΔCat.
Artifact	24	+0.72	13/24	0.00
Incident	24	+0.83	17/24	-0.17
Data	24	+1.13	18/24	-0.35
Config	24	+1.06	22/24	-0.36
Evidence	24	+0.98	21/24	-0.51
Resource	24	+1.82	19/24	-0.54

Table 5: Case-family analysis across four model pairs. Win counts matched cells where TRACE-EVO has a higher held-out score than Static-Growth.

Models	ΔCat.	Org.	Trial	Com.	Roll.
Dola Seed	-0.44	2.33	0.78	0.44	0.33
DeepSeek	-0.42	4.53	1.28	0.39	0.89
MiniMax	-0.15	6.36	1.53	0.03	1.50
GPT-5.5/Seed	-0.28	1.94	0.64	0.36	0.28

Table 6: Cost and reliability of TRACE-EVO across model pairs. ΔCat. compares catastrophic failure rate with Static-Growth; lower Org., Trial, Com., and Roll. indicate fewer organization actions.

Case-Family Analysis We further group the paired cells by the six TRACEBench task families. Table 5 shows the largest gain on resource operations, with consistent improvements on data, configuration, and evidence tasks where changing state and stale role-local knowledge create downstream risk. Artifact cases form the weakest family, but the average held-out gain remains positive.

Cost and Reliability Table 6 reports the reliability cost of role evolution. TRACE-EVO reduces catastrophic failures in every completed model pair, but the required organization actions differ by model family. MiniMax has the highest rollback rate and the smallest failure reduction, matching its weaker average score gain in the main results.

Conclusion

We introduced TRACE-EVO, a framework for contribution-driven role reassignment and governed knowledge transfer in long-horizon heterogeneous multi-agent systems with fixed model identities. We also built TRACEBench, which freezes teams after growth-phase work to test held-out organizational transfer. Across multiple model pairs, TRACE-EVO improves over static organizations and reduces catastrophic failures, with gains driven by capability-aware matching and controlled memory and skill transfer. These results show that long-horizon agent teams should be evaluated as evolving organizations, not only as collections of prompts or independent agents.

References

Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Chan, C.-M.; Yu, H.; Lu, Y.; Hung, Y.-H.; Qian, C.; Qin, Y.; Cong, X.; Xie, R.; Liu, Z.; Sun, M.; and Zhou, J. 2024. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *The Twelfth International Conference on Learning Representations*.

Drouin, A.; Gasse, M.; Caccia, M.; Laradji, I. H.; Del Verme, M.; Marty, T.; Vazquez, D.; Chapados, N.; and Lacoste, A. 2024. WorkArena: How Capable Are Web Agents at Solving Common Knowledge Work Tasks? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 11642–11662. PMLR.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2974–2982. AAAI Press.

Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; Ran, C.; Xiao, L.; Wu, C.; and Schmidhuber, J. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*.

Huang, Q.; Vora, J.; Liang, P.; and Leskovec, J. 2024. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 20271–20309. PMLR.

Jimenez, C. E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; and Narasimhan, K. R. 2024. SWE-bench: Can Language Models Resolve Real-world Github Issues? In *The Twelfth International Conference on Learning Representations*.

Li, G.; Hammoud, H. A. A. K.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Advances in Neural Information Processing Systems*.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; Zhang, S.; Deng, X.; Zeng, A.; Du, Z.; Zhang, C.; Shen, S.; Zhang, T.; Su, Y.; Sun, H.; Huang, M.; Dong, Y.; and Tang, J. 2024a. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*.

Liu, Z.; Zhang, Y.; Li, P.; Liu, Y.; and Yang, D. 2024b. A Dynamic LLM-Powered Agent Network for Task-Oriented Agent Collaboration. In *First Conference on Language Modeling*.

Mialon, G.; Fourrier, C.; Wolf, T.; LeCun, Y.; and Scialom, T. 2024. GAIA: A Benchmark for General AI Assistants. In *The Twelfth International Conference on Learning Representations*.

Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S. G.; Stoica, I.; and Gonzalez, J. E. 2024. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery.

Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; Xu, J.; Li, D.; Liu, Z.; and Sun, M. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15174–15186. Bangkok, Thailand: Association for Computational Linguistics.

Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; Zhao, S.; Hong, L.; Tian, R.; Xie, R.; Zhou, J.; Gerstein, M.; Li, D.; Liu, Z.; and Sun, M. 2024. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *The Twelfth International Conference on Learning Representations*.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J. N.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 4295–4304. PMLR.

Shapley, L. S. 1953. A Value for n-Person Games. In *Contributions to the Theory of Games*, volume 2, 307–317. Princeton University Press.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2018. Value-Decomposition Networks for Cooperative Multi-Agent Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*.

Wang, Z.; Wang, Y.; Liu, X.; Ding, L.; Zhang, M.; Liu, J.; and Zhang, M. 2025. AgentDropout: Dynamic Agent Elimination for Token-Efficient and High-Performance LLM-Based Multi-Agent Collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 24013–24035. Vienna, Austria: Association for Computational Linguistics.

Wolpert, D. H.; and Tumer, K. 2002. Collective Intelligence, Data Routing and Braess' Paradox. *Journal of Artificial Intelligence Research*, 16: 359–387.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; Awadallah, A. H.; White, R. W.; Burger, D.; and Wang, C. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *First Conference on Language Modeling*.

Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T. J.; Cheng, Z.; Shin, D.; Lei, F.; Liu, Y.; Xu, Y.; Zhou, S.;

Savarese, S.; Xiong, C.; Zhong, V.; and Yu, T. 2024. OS-World: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems*.

Xu, F. F.; Song, Y.; Li, B.; Tang, Y.; Jain, K.; Bao, M.; Wang, Z.; Zhou, X.; Guo, Z.; Cao, M.; Yang, M.; Lu, H. Y.; Martin, A.; Su, Z.; Maben, L.; Mehta, R.; Chi, W.; Jang, L.; Xie, Y.; Zhou, S.; and Neubig, G. 2025. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. In Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; and Chen, N., eds., *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc.

Yang, J.; Jimenez, C. E.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K.; and Press, O. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems*.

Yao, S.; Shinn, N.; Razavi, P.; and Narasimhan, K. R. 2025. {τ}-bench: A Benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

Zhang, J.; Xiang, J.; Yu, Z.; Teng, F.; Chen, X.-H.; Chen, J.; Zhuge, M.; Cheng, X.; Hong, S.; Wang, J.; Zheng, B.; Liu, B.; Luo, Y.; and Wu, C. 2025. AFlow: Automating Agentic Workflow Generation. In *The Thirteenth International Conference on Learning Representations*.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. MemoryBank: Enhancing Large Language Models with Long-Term Memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17): 19724–19731.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; Alon, U.; and Neubig, G. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*.

Zhuge, M.; Wang, W.; Kirsch, L.; Faccio, F.; Khizbullin, D.; and Schmidhuber, J. 2024. GPTSwarm: Language Agents as Optimizable Graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 62743–62767. PMLR.