Collaborative Multi-Agent Test-Time Reinforcement Learning for Reasoning

Anonymous ACL submission

Abstract

Multi-agent systems have evolved into practical LLM-driven collaborators for many applications, gaining robustness from diversity and cross-checking. However, multi-agent RL (MARL) training is resource-intensive and unstable: co-adapting teammates induce non-stationarity, and rewards are often sparse and high-variance. Therefore, we introduce **Multi-Agent Test-Time Reinforcement Learning (MATTRL)**, a framework that injects structured textual experience into multi-agent deliberation at inference time. MATTRL forms a multi-expert team of specialists for multi-turn discussions, retrieves and integrates test-time experiences, and reaches consensus for final decision-making. We also study credit assignment for constructing a turn-level experience pool, then reinjecting it into the dialogue. Across challenging benchmarks in medicine, math, and education, MATTRL improves accuracy by an average of 3.67% over a multi-agent baseline, and by 8.67% over comparable single-agent baselines. Ablation studies examine different credit-assignment schemes and provide a detailed comparison of how they affect training outcomes. MATTRL offers a stable, effective and efficient path to distribution-shift-robust multi-agent reasoning without tuning. Code is in Software part under submission page.

1 Introduction

Multi-agent systems have moved from early algorithmic prototypes to practical LLM-driven collaborators. Across math, coding, web interaction, and analytical benchmarks, these multi-agent systems reliably outperform comparable single-agent baselines, as diversity and cross-checking improve robustness under distribution shift.

Recent works explore collaborative multi-agent frameworks to enhance LLM agents' capabilities. For example, AutoGen (Wu et al., 2024) (orchestrated multi-agent dialogues with tool use and human-in-the-loop), CAMEL (Li et al., 2023) (role-playing with inception prompting), AgentVerse (Chen et al., 2023) (an open platform for cooperative problem solving and social simulation), ChatDev (Qian et al., 2023) (specialized software agents for design, coding, and testing), and Magentic-One (Fourney et al., 2024) (an orchestrator that routes tasks among specialized agents for web/local workflows). In parallel, the success of DeepSeek-R1 (Guo et al., 2025) has catalyzed reinforcement learning (RL) as a post-training paradigm for stronger reasoning. Efforts to extend RL to the multi-agent setting include MAPoRL (Park et al., 2025), which jointly optimizes multi-model discussions and final answers via RL, and ReMA (Wan et al., 2025), which separates high-level meta-thinking from low-level reasoning into two agents and trains them with GRPO.

However, MARL remains resource-intensive and can erode general abilities when adapted to a single domain. Training stability is also difficult to guarantee due to (i) non-stationarity from simultaneously evolving teammates, which shifts state and return distributions, and (ii) sparse, high-variance rewards. Hence, we propose Multi-Agent Test-Time Reinforcement Learning (MATTRL), an adaptation framework that injects test-time textual experience into the collaborative process. Instead of updating weights, MATTRL conditions behavior with structured experience, enabling rapid, distribution-shift-robust adaptation to new tasks/domains without harming original generality. Additionally, textual experience provides richer turn-level signals about collaboration quality and reasoning than scalar rewards alone. Textual experience mitigates key MARL pain points by keeping policies fixed and providing dense, stepwise experience at every turn.

The crucial components of MATTRL include (1) various group-to-agent credit assignment strategies for experience selection, (2) construction of an experience pool from test time examples, and (3)

integration of the experience pool into the multi-agent collaborative process. MATTRL first instantiates a team of specialized agents. The agents deliberate in multi-turn discussions, drawing on relevant prior experience to aggregate evidence and move toward agreement. The process terminates when agreement is reached or a predefined turn limit is met. A designated coordinator agent then summarizes the discussion, consolidates the accumulated evidence, and outputs the final decision. To retrieve experience, each agent utterance is first scored using both individual-performance signals and a decayed terminal shared reward. For constructing the experience pool, high-scoring utterances are distilled into textual experiences and added to the pool for subsequent retrieval and integration. Experiments show that, on benchmarks spanning medicine, math, and education, MATTRL boosts average performance by 3.67% over the multi-agent framework and by 8.67% over comparable single-agent baselines. Furthermore, we systematically explored multiple credit-assignment schemes for group-credit attribution in experience selection, ranging from naïve shared credit to difference rewards and Shapley-style approximations. To summarize, our contributions focus on these three perspectives:

- We propose the first Multi-agent Test Time Reinforcement Learning framework, MATTRL, leveraging textual experience to enhance the multi-agent system.
- We further validate the effect of different credit assignments on experience construction and the final decision.
- Experiments conducted on medical, math and education benchmarks achieve a new SOTA performance based on MATTRL.

2 Related Work

LLM-based multi-agent collaboration. Recent advancements in LLM-based multi-agent systems have emphasized scalable collaboration mechanisms for complex task-solving. Surveys (Tran et al., 2025) outline key coordination strategies in LLM-driven multi-agent systems, enabling groups of agents to work collectively at scale. MacNet (Qian et al., 2024) explores the benefits of continuously adding agents to enhance performance in collaborative settings. Multi-agent systems utilizing LLMs also emerge as tools for enhancing medical decision-making processes. MDAgents (Kim et al., 2024) introduces adaptive collaboration among LLMs to address gaps in clinical reasoning and diagnostics. Multi-agent conversational framework, MAC(Chen et al., 2025) boost diagnostic accuracy through interactive agent dialogues.

Reinforcement learning for LLM reasoning. Reinforcement learning techniques have been increasingly applied to refine reasoning capabilities in large language models. Models such as DeepSeek-R1 (Guo et al., 2025) demonstrate RL's potential to enhance LLM reasoning without relying on human-annotated data. Recent work also systematize RL for reasoning-centric LLMs. SimpleRL-Zoo (Zeng et al., 2025) conducts a broad, controlled study of RL on open-base models, showing that careful reward formatting and difficulty curation drive reliable gains across benchmarks. Understanding R1-Zero-Like Training (Liu et al., 2025) disentangles base-model priors from optimizer effects, identifies length-inducing biases in GRPO, and introduces a debiased variant (Dr.GRPO) that yields strong math results with lightweight recipes. Complementing these, Beyond "Aha!" (Hu et al., 2025b) aligns meta-abilities explicitly, spanning deductive, inductive, and abductive skills, via automatically verifiable tasks and targeted RL, achieving consistent improvements over instruction-tuned baselines.

Test-time adaptation and structured experience. Test-time adaptation methods allow LLMs to dynamically adjust to new domains during inference without additional training. The Test-Time Learning (TTL) paradigm, such as TLM (Hu et al., 2025a), adapts models using only unlabeled test data to handle domain shifts effectively. Test-Time Reinforcement Learning (TTRL) (Zuo et al., 2025) converts test-time scaling signals into pseudo-rewards to train LLMs on unlabeled data, enabling self-evolution and substantial gains. Study (Wang et al., 2025) also evaluate LLM improvements from structured experience using semantic games as testbeds resistant to saturation.

Credit assignment under collaboration. Credit assignment in multi-agent collaborations involving LLMs tackles the challenge of fairly attributing contributions in cooperative settings. LLM-based methods reformulate credit assignment as pattern recognition to achieve efficient and effective distribution in Multi-agent system. Approaches like Shapley-Coop (Hua et al., 2025) address emergent

cooperation in self-interested multi-agent systems through value-based credit allocation. Frameworks such as LLM-MCA (Nagpal et al., 2025) utilize large language models for multi-agent credit assignment in reinforcement learning contexts. Systems like CollabUIAgents (He et al.) advance multi-agent learning by incorporating LLM-guided credit re-assignment and synthetic preference data.

3 Methodology

3.1 Multi-Expert Team Collaboration

We study a general multi-agent decision-making setting. Each instance provides: (i) a task record (or user context) $\mathcal{X}$, (ii) a coordinator agent $\mathrm{LLM}_{\mathrm{Coo}}$, (iii) an expert catalog $\mathcal{SP}$ (a pool of specialist agents with textual expertise descriptions), and (iv) a callable test-time experience pool $\mathcal{E}$ (Sec. 3.2). At test time, $\mathrm{LLM}_{\mathrm{Coo}}$ optionally retrieves relevant experiences to strengthen the current decision. The expert-team consultation follows three stages with a preset maximum of $R_{\max}$ discussion rounds. Our hospital consultation experiments are a concrete instantiation by interpreting $\mathcal{X}$ as a patient record and $\mathcal{SP}$ as clinical departments.

Stage I: Team formation. Rather than letting LLMs freely invent roles, we select an expert team $\mathrm{TEAM} \subseteq \mathcal{SP}$ based on the task record $\mathcal{X}$ using a recruitment prompt (Appendix A.4) that conditions on $\mathcal{X}$ and each specialist's expertise description:

$$\mathrm{TEAM} \leftarrow \mathrm{LLM}_{\mathrm{Coo}}(\mathcal{X}, \mathcal{SP}). \tag{1}$$

Each specialist $s \in \mathrm{TEAM}$ maintains a round-indexed opinion set $\mathcal{O}_s^{(r)}(\mathcal{X})$ and a convergence flag $f_s^c \in \{\mathrm{False}, \mathrm{True}\}$ (initialized to False). We denote the team union at round r as

$$\mathcal{O}^{(r)}(\mathcal{X}) = \bigcup_{s \in \mathrm{TEAM}} \mathcal{O}_s^{(r)}(\mathcal{X}). \tag{2}$$

Stage II: Consensus via experience-augmented dialogue. The team proceeds in synchronized rounds $r = 0, 1, \ldots, R_{\max}$. In each round, each non-converged specialist s retrieves task-relevant experiences and then issues a revised opinion.

We denote the retrieved experience set for s as

$$\mathrm{ER}_s \leftarrow \mathrm{Retrieve}\big(\mathcal{E}\,;\, \mathcal{X}, u_s^{(r)}\big), \tag{3}$$

where $u_s^{(r)}$ is the current utterance/contextual query formed by specialist s at round r. In our implementation, $\mathrm{Retrieve}(\cdot)$ uses a shared encoder $f(\cdot)$ (`Qwen3-Embedding-4B` (Zhang et al., 2025)) and a `FAISS` index (Douze et al., 2024) to select top-K entries by cosine similarity. Details are in Appendix A.7, B.9 and C.1.1. The retrieved entries are appended to the prompt under a fixed template.

The specialist then updates its opinion conditioned on its previous state and retrieved evidence:

$$\mathcal{O}_s^{(r)}(\mathcal{X}) \leftarrow \mathrm{LLM}_s\big(\mathcal{X},\, \mathcal{O}_s^{(r-1)}(\mathcal{X}),\, \mathrm{ER}_s\big). \tag{4}$$

We define the incremental update as

$$\Delta\mathcal{O}_s^{(r)} := \mathcal{O}_s^{(r)}(\mathcal{X}) \setminus \mathcal{O}_s^{(r-1)}(\mathcal{X}). \tag{5}$$

Opinions are then synchronized in a meeting step that shares salient updates with all members. Specifically, $\mathrm{MEETING}(\cdot)$ is a lightweight aggregation operator that takes all specialists' incremental updates $\{\Delta\mathcal{O}_s^{(r)}\}_{s\in\mathrm{TEAM}}$ and produces a deduplicated, concise shared bulletin $\Delta\mathcal{O}_{\mathrm{share}}^{(r)}$:

$$\Delta\mathcal{O}_{\mathrm{share}}^{(r)} \leftarrow \mathrm{MEETING}\Big(\{\Delta\mathcal{O}_s^{(r)}\}_{s\in\mathrm{TEAM}}\Big). \tag{6}$$

Each specialist receives $\Delta\mathcal{O}_{\mathrm{share}}^{(r)}$ in the next round's context to align beliefs and avoid redundant discussion. Each specialist receives $\Delta\mathcal{O}_{\mathrm{share}}^{(r)}$ in the next round's context. A specialist is marked converged when no further changes are proposed, i.e., $\Delta\mathcal{O}_s^{(r)} = \varnothing$. The process halts when all specialists converge or when $r = R_{\max}$.

Stage III: Report synthesis and final decision. After the bounded discussion, the coordinator agent synthesizes the team's cumulative evidence into a discussion report DR:

$$\mathrm{DR} = \mathrm{SUMMARY}\left[\bigcup_{r=0}^{R_{\max}} \bigcup_{s\in\mathrm{TEAM}} \mathcal{O}_s^{(r)}(\mathcal{X})\right]. \tag{7}$$

The coordinator agent may also perform its own retrieval ER from $\mathcal{E}(\mathcal{X})$, and outputs the final decision A conditioned on the task record and aggregated evidence:

$$A \leftarrow \mathrm{LLM}_{\mathrm{Coo}}\big(\mathcal{X},\, \mathrm{DR},\, \mathrm{ER}\big). \tag{8}$$

Remarks. Stage I grounds role selection in a predefined expert catalog $\mathcal{SP}$, Stage II enforces a bounded multi-turn consensus process with explicit convergence checks and retrieval-augmented evidence, and Stage III separates evidence aggregation (report synthesis) from decision making, improving controllability and auditability.

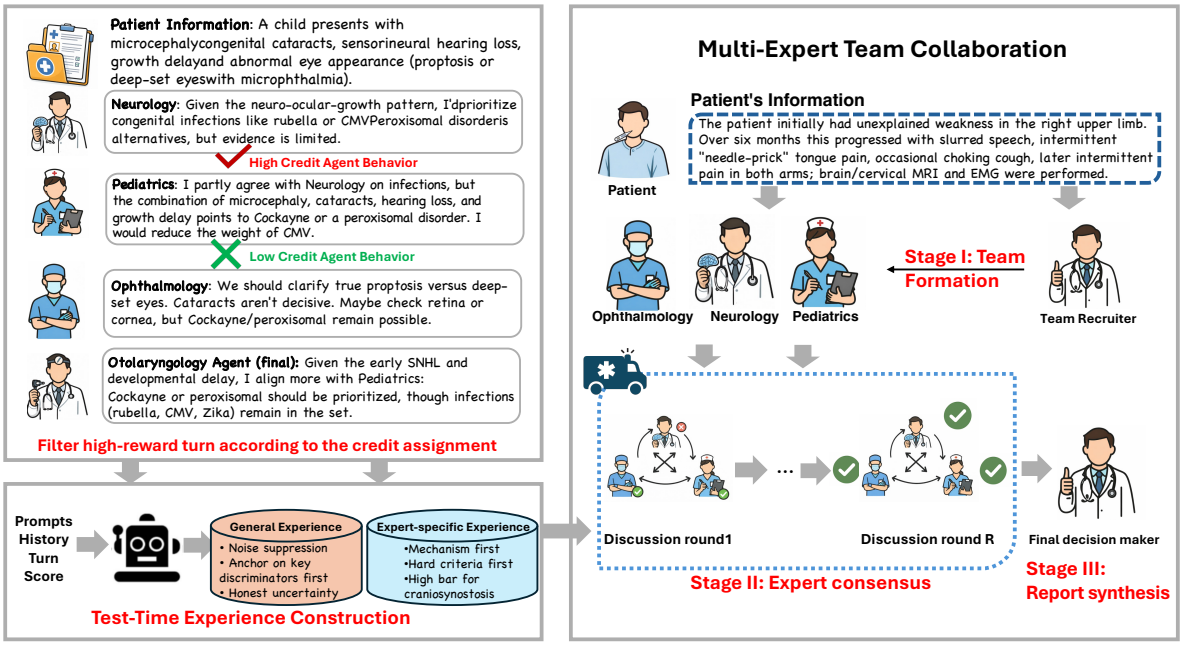

Figure 1: MATTRL overview. The figure uses **medical diagnosis** as a running example, but the framework is domain-general. **Math** and **education** instantiations are in Appendix B.1 and C.1.

3.2 Test-Time Experience Construction

Given a multi-agent transcript with R turns, let TEAM denote the set of specialist agents. At turn $t \in \{1, \ldots, R\}$, agent $i \in$ TEAM produces an utterance $u_{i,t}$ under its observable context/history $\mathcal{H}_{i,t}$. We employ an LLM judge (rubrics in Appendix A.5, B.6, and C.4.6) to evaluate each utterance along domain-relevant axes (e.g., correctness, information gain, relevance to the task, clarity, *etc.*), yielding an *individual score*:

$$s_{i,t} \;=\; \phi_{\text{LLM}}\big(u_{i,t},\, \mathcal{H}_{i,t};\, \text{Rubric}\big) \in [0,1]. \qquad (9)$$

Contribution ratio and terminal shared reward. Assume we obtain a single *terminal* team-level outcome score G at the end of the consultation (e.g., task success), where $G \in [0,1]$. Let R be the actual number of turns (with $R \le R_{\max}$). We allocate G back to each turn via a decay kernel and split each turn's share across agents by contribution ratios. Define per-turn decay weights

$$w_t \;=\; \gamma^{R-t} \qquad (10)$$

The later turns receive higher weight when $\gamma < 1$. Each agent's contribution ratio $c_{i,t}$ is estimated by proportional normalization of per-agent scores within each turn:

$$c_{i,t} = \frac{s_{i,t}}{\sum_{j \in \text{TEAM}} s_{j,t} + \epsilon}, \qquad s_{i,t} \ge 0, \qquad (11)$$

where ϵ avoids division by zero.

Turn-level reward for each agent. We fuse individual and terminal team signals:

$$r_{i,t} \;=\; \lambda\, s_{i,t} \;+\; (1-\lambda)\, G \cdot w_t \cdot c_{i,t}, \qquad \lambda \in [0,1]. \qquad (12)$$

Selection of high-value utterances. To construct reusable test-time experiences, we select high-value snippets using a threshold:

$$\mathcal{I}_i^{\text{keep}} = \{\, t \mid r_{i,t} \ge \tau \,\}. \qquad (13)$$

From high-scoring utterances to textual experience. For each $(i,t) \in \mathcal{I}_i^{\text{keep}}$, we map the context $\mathcal{H}_{i,t}$, utterance $u_{i,t}$, and quantitative signals $r_{i,t}$ into a structured, retrievable *textual experience entry* using an LLM summarizer (prompt templates in Appendix A.6):

$$e_{i,t} \;=\; \Psi_{\text{LLM}}\Big(\mathcal{H}_{i,t},\, u_{i,t},\, r_{i,t};\, \text{Template}_{\text{exp}}\Big). \qquad (14)$$

This yields a test-time experience pool

$$\mathcal{E} \;=\; \{\, e_{i,t} \mid i \in \text{TEAM},\; t \in \mathcal{I}_i^{\text{keep}} \,\}, \qquad (15)$$

We define a *textual experience entry* as a compact, structured text record that is easy to retrieve and reuse. Each entry stores (i) minimal task context for retrieval, (ii) the actionable step taken, and (iii) a short rationale for the assigned credit.

Method	Hit@1	Hit@3	Hit@5	Hit@10	MRR
MDAgent	0.32	0.49	0.57	0.68	0.46
RareAgents	0.29	0.38	0.47	0.68	0.42
RareAgent-Refined	0.35	0.49	0.57	0.70	0.47
MATTRL	**0.39**	**0.51**	**0.61**	**0.75**	**0.51**

Table 1: Experimental Results on Baselines and MATTRL for medicine benchmark

4 Experiments

4.1 Setup

Datasets and Domain Settings In **Medicine** setting, RareBench (Chen et al., 2024b) evaluates LLMs as rare-disease specialists across four tasks. We focus on Task 4 (differential diagnosis among universal rare diseases) with 2,185 cases covering 421 diseases, and cast the task as a multi-agent consultation: an attending agent orchestrates domain specialists to independently propose and justify differential diagnoses from the patient record, critique peers' evidence, and iteratively refine toward a consensus shortlist. **Math**: We utilize HLE (Humanity's Last Exam) (Phan et al., 2025) with text-only math problems (856 samples), a challenging benchmark of expert-level questions, to assess collaborative problem solving. We report exact-match solve rate via LLM judgement and quantify the improvement brought by multi-agent deliberation with test time experience. **Education**:We study teaching-oriented interaction with a three-stage designs: pre-test, instruction, and post-test. The student first answers with reasoning. Then a teacher, given the question, gold answer, and the student's response, conducts a two-round teaching dialogue. Finally, the student re-answers. We sample 300 questions from SuperGPQA (Du et al., 2025) with GPT-4o as the student and GPT-5 as the teacher, and measure learning gains by post-test accuracy improvement. We also demonstrate the detailed examples, settings and prompts for these three domains in Appendix A, B and C.

Baselines. In **medicine** settings, We compare against two agentic baselines. *MDAgents* (Kim et al., 2024) is an adaptive collaboration framework that estimates case complexity, recruits an appropriate team, performs multi-turn analysis–synthesis, and ends with moderator review. Its dynamic structure and moderation/knowledge components improve medical QA and diagnosis. RareAgents (Chen et al., 2024a) targets rare-disease diagnosis via a patient-centered Multi-disciplinary Team (MDT) with specialist orchetration, case-memory retrieval, and tool use. Since its memory corpus and tool library are not released, we reimplement the MDT-only version. We also introduce *RareAgents-Refined*, a prompt-engineered variant that enforces role-focused, critical peer review and discourages fabricated tests/results, reducing confirmation bias and hallucinations and yielding consistent gains. For **math** and **education** domains, we use a **single-agent** solver/teacher that directly performs the task as one baseline. We then compare it against our **multi-agent** instantiation described in Section 3.1, where multiple experts independently propose, critique, and iteratively refine solutions (or teaching moves) with periodic synchronization/aggregation. This isolates the effect of test-time experience.

Metrics **Medicine.** We report *Hit@k* and *MRR* on the attending agent's *final ranked differential list/shortlist*, where Hit@k is the fraction of cases whose ground-truth disease appears within the top-k predictions, and MRR averages $1/\text{rank}$ of the correct disease. Higher is better. **Math.** We report exact-match solve rate (*Acc*), where a problem is counted as solved if the final answer matches the reference under an LLM judge. **Education (SuperGPQA).** We measure learning by pre-test and post-test accuracy and report learning gains as $\Delta Acc = Acc_{\text{post}} - Acc_{\text{pre}}$ (higher indicates stronger instructional improvement).

Paremeters Settings We use GPT-5 (OpenAI, 2025) as the backbone model is our MATTRL framework and other aforementioned LLMs are also GPT-5. The number of experts is 3, and the maximum conversation turns are limited to 3. For experience text construction, we select 30 cases. For all utterance from agents, we extract the Top 25% scored records for further construction.

4.2 Results

As demonstrated in Table 1, in medicine task, MATTRL achieves the strongest overall retrieval quality. Averaged over k = 1, 3, 5, and 10, its Hit@k is 0.565, higher than MDAgent at 0.515 and RareAgents-Refined at 0.528, and it also attains

Method	Single Agent	Multi-Agent	MATTRL
Acc	0.27	0.33 $_{(+0.06)}$	0.36 $_{(+0.09)}$

Table 2: **Math (Accuracy Comparison with Per-Method Improvement).** We report exact-match accuracy on HLE math problems. Numbers in the bottom-right indicate the absolute change in accuracy relative to the single agent baseline

the highest MRR of 0.51. The most pronounced advantages appear at Hit@1, indicating better top-rank precision, and at Hit@10, indicating more reliable shortlist coverage. Overall, the results suggest that test-time collaborative adaptation yields benefits beyond those achievable through prompt optimization alone.

As shown in Table 2, the single-agent baseline achieves an exact-match accuracy of 0.27 on HLE. Introducing multi-agent deliberation improves performance to 0.33, indicating a modest benefit from parallel proposal and critique. MATTRL yields a larger gain, reaching 0.36, suggesting that test-time experience further strengthens collaborative problem solving beyond deliberation alone.

For Education, as shown in Table 3, all methods start from the same pre-test accuracy ($Acc_{\text{pre}} = 0.44$), ensuring a controlled comparison where improvements reflect instructional effectiveness rather than initial student performance. The single-agent teacher increases accuracy to $Acc_{\text{post}} = 0.60$ ($\Delta Acc = 0.16$). Replacing it with a multi-agent teacher that proposes and critiques teaching moves yields a much larger gain, suggesting that deliberation helps identify misconceptions and select more effective explanations. MATTRL further achieves the best post-test performance at $Acc_{\text{post}} = 0.77$ with the highest learning gain ($\Delta Acc = 0.33$), nearly doubling the improvement of the single-agent baseline. Overall, the results indicate that collaboration substantially enhances teaching outcomes, and test-time experience provides additional benefits beyond collaboration alone.

Method	Acc_{pre}	Acc_{post}	ΔAcc
Single Agent	0.44	0.60	0.16
Multi-Agent	0.44	0.73	0.29
MATTRL	0.44	0.77	0.33

Table 3: **Education (Learning Gains in a Pre-test → Tutoring → Post-test Setup).** We report pre-test accuracy (Acc_{pre}), post-test accuracy (Acc_{post}), and learning gain ($\Delta Acc = Acc_{\text{post}} - Acc_{\text{pre}}$).

5 Analysis

All ablations and analysis conducted below are based on medicien dataset (RareBench).

5.1 Group-to-Agent Credit Assignment

We compare naive averaging, Difference Rewards, and Shapley-style approximations for attributing team returns at each turn to individual agents.

As we mentioned in section 3.2, We compute agent-specific *credit scores* $q_{i,t}$ for agent i at turn t and map them to contribution ratios via a shared normalization to ensure comparability:

$$c_{i,t} \;=\; \frac{\exp(\beta\, q_{i,t})}{\sum_{j\in\text{TEAM}} \exp(\beta\, q_{j,t})} \qquad \beta > 0. \quad (16)$$

Difference Rewards. For agent i at turn t, define the counterfactual where i is neutralized while others remain:

$$q_{i,t}^{\text{Diff}} \;=\; F_t(\text{TEAM}) - F_t(\text{TEAM} \setminus \{i\}) \quad (17)$$

where $F_t(\cdot)$ is the turn-t team objective (e.g., consensus gain or hypothesis-space reduction). In practice, $F_t(\text{MDT} \setminus \{i\})$ is approximated by rerunning the turn with i's utterance replaced by a no-op, or via a learned proxy (Appendix).

Shapley-style approximations. The Shapley value averages i's marginal effect across orders:

$$q_{i,t}^{\text{Shap}} \;=\; \mathbb{E}_{\pi}\big[F_t\big(S_{\pi}^{<i} \cup \{i\}\big) - F_t\big(S_{\pi}^{<i}\big)\big] \quad (18)$$

with $S_{\pi}^{<i}$ the set of agents preceding i in permutation π. We estimate $q_{i,t}^{\text{Shap}}$ via K Monte Carlo permutations (or small-coalition sampling) with cached $F_t(\cdot)$ to control cost. Unless stated otherwise, all schemes use the same $F_t(\cdot)$ and the same normalization (identical β) before feeding $c_{i,t}$ into the decay-weighted terminal allocation in *Contribution ratio and terminal shared reward.*

Method	Hit@1	Hit@3	Hit@5	Hit@10
Naive	0.39	0.51	0.61	0.75
Difference	0.40	0.53	0.61	0.74
Shapley	0.35	0.49	0.59	0.75

Table 4: Performance comparison among different credit assignments for experience construction. Naive represents the Naive method we mentioned in section 3.2, Difference denotes the Difference Rewards and Shapley is Shapley-style approximations.

As shown in Table 4, DIFFERENCE yields the best strict-precision performance (Hit@1/3 = 0.40/0.53), outperforming NAIVE (0.39/0.51) and

SHAPLEY (0.35/0.49). At broader cutoffs the methods are similar: Hit@5 is tied for DIFFERENCE/NAIVE (0.61) and Hit@10 is nearly identical (0.74–0.75). We attribute DIFFERENCE's gains on tight metrics to reduced free-riding noise: contrasting the full team with a counterfactual where agent i is neutralized better isolates decisive turns and produces sharper credit peaks after normalization. By contrast, SHAPLEY tends to spread credit across coalitions (and is variance-prone under limited permutations), which dilutes peaks and hurts Hit@1/3 despite comparable Hit@10.

Why Shapley underperforms. We observe that Shapley-style selection tends to reward *peer-review/alignment behaviors* that improve coherence and consensus but have limited influence on the decisive inference steps. Since Shapley averages marginal effects across many coalitions, sharp decision moves are diluted while low-variance meta-behaviors accumulate steady credit (e.g., "integrate peer comments coherently," "maintain cross-specialty consensus"). By contrast, Naive more often elevates *decision-centric hints* with short feedback loops because it ties credit to single-run outcome deltas (e.g., "prioritize MMA over PA when biomarkers dominate," "merge weakly anchored subtypes into a low-priority bucket"), yielding sharper hypothesis ranking and stronger top-rank accuracy. Beyond hit rates, compute and stability also favor Difference. Shapley needs many marginal evaluations and has higher estimator variance unless heavily sampled; Naive is cheapest but sensitive to correlated noise. Difference offers a practical middle ground with one counterfactual per agent, providing a low-variance, high-leverage signal at modest cost. Overall, we recommend Shapley when fairness is paramount and budget allows, Naive as a low-cost baseline, and Difference as the default when precision and efficiency matter.

5.2 Adaptive collaboration between single agent and multi-agent framework

To further improve the practicality of MATTRL, we additionally compare against a single-agent baseline using chain-of-thought (CoT) reasoning and develop an Adaptive method that learns to route each case to either the single agent or MATTRL. The classifier makes the routing decision based on features capturing symptom complexity, need for multidisciplinary consultation, number of specialties involved, cross-specialty divergence, and risk of single-expert misguidance. As shown in Table 5, the single-agent CoT baseline is already strong, and the Adaptive router further improves performance, achieving average gains of 10% over the single agent and 5.5% over MATTRL.

Method	Hit@1	Hit@3	Hit@5	Hit@10
Single-Agent	0.39	0.49	0.56	0.64
MATTRL	0.39	0.51	0.61	0.75
Adaptive	**0.45**	**0.58**	**0.66**	**0.79**

Table 5: Results of Single-Agent, MATTRL, and Adaptive Router (Adaptive in below table).

Single-agent excels when cases show standardized diagnostic "fingerprints" that a one-shot integration can resolve, evidence is concentrated in one specialty, and the task prioritizes internal consistency with a concise explanation. Multi-agent is stronger when evidence spans multiple specialties or modalities and needs cross-validation, the goal extends to risk assessment/care planning/test prioritization, and the task benefits from systematic counterfactuals and competing hypotheses for robust differentials. This aligns with our analysis for the classifier in adaptive method and the error analysis for both single agent and MATTRL.

Our classifier routed 282 cases to the single-agent solver and 840 to MATTRL. Empirically, many instances that are internally consistent are solvable by the single agent, yet the multi-agent discussion can introduce noise that harms accuracy on those same cases. A Venn-style breakdown of correctness shows: Only the single agent solves around 300 cases, only MATTRL solves 400+ cases, and both solve 357 cases.

5.3 Scaling with Team Size

We study how performance scales as the number of collaborating agents increases (e.g., 1, 3, 7, 9). As shown in Figure 2, increasing the number of agents does not uniformly improve performance. For Hit@1, accuracy peaks at three agents and then declines as the team grows. Because Hit@1 requires strict precision, larger teams introduce more divergent opinions and make consensus harder to reach. In contrast, Hit@3 and Hit@5 exhibit modest, steady gains with scale. Hit@10 benefits the most from scaling, as broader discussions surface more plausible candidates and are more tolerant to noise. Notably, a three-agent team outperforms a single agent by about 14% on Hit@10. Practically, smaller teams (e.g., three agents) are preferable

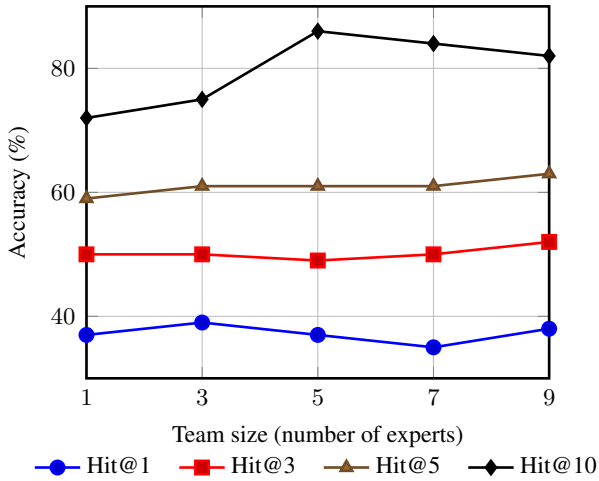

Figure 2: GPT-5 Multi-Agent: Acc. by Team Size.

for high-precision decisions, whereas larger teams help when broader recall is desired.

Experiences

General Experience:

Noise suppression: Oppose the vague "as long as it matches" alignment. Broad statements like "can fit multiple diseases / seems consistent" dilute discriminative power and prolong an unhelpful tail. The requirement is to clearly explain the mechanistic consistency from feature → diagnosis; otherwise, down-weight or exclude it.

Anchor on key discriminators first: Use the few features that "most sharply split candidate diseases in one cut" and are observed across multiple specialties as the main backbone (main sequence) for ranking. Other evidence should only serve as fine-tuning, not overturn the backbone.

Honest uncertainty: When the evidence is insufficient to make a unique choice, explicitly state "insufficient to select a single diagnosis" to avoid over-commitment; ranking must stay constrained by evidence.

Disease-specific Experience:

Mechanistic clarification before assumptions: For the cause of leukocoria ("white pupil"), attribution (lens vs retina) must be clarified first; if evidence is lacking, state "insufficient for attribution." This prevents prematurely over-weighting a certain subtype/spectrum.

Let hard criteria dominate near-neighbor ordering: For axial/epiphyseal hard features like odontoid hypoplasia or delayed ossification, require an explicit explanation of how they shift the relative positions of close candidates. Within related spectra, use these high-weight hard indicators as tie-breakers, rather than vague impressions.

Set a high evidence threshold for the craniosynostosis spectrum: Non-specific morphologies such as shallow orbits/proptosis, thick skull, etc., are insufficient to raise its weight. Without hallmark evidence of suture involvement, keep it low-ranked and state this explicitly. Also, remind to explicitly address cranial signs that contradict the main hypothesis, to avoid anchoring bias.

Figure 3: General & disease-specific experience

5.4 Experience Examples

Figure 6 shows two kinds of reusable test-time experiences that MATTRL extracts from consultation transcripts. *General experiences* are cross-disease rules that improve discriminability and keep discussion disciplined. For instance, they require mechanism-grounded justifications instead of vague "seems consistent", prioritize a small set of high-yield discriminators as the ranking backbone, and state uncertainty explicitly when evidence is weak. *Disease-specific experiences* are concise, concrete checks that guide fine-grained ordering among close candidates (e.g., first clarify the locus of leukocoria before assuming a subtype; let high-weight skeletal markers adjust relative ranks; keep craniosynostosis low without direct evidence of suture involvement). Practically, we select utterances with higher reward via credit assignment, distill their underlying rationale into brief, textual experience snippets, and retrieve them at inference to stabilize multi-agent deliberation and improve accuracy without updating model weights.

5.5 Few-shot vs. Test-time Experience

To test whether MATTRL's gains stem merely from supplying extra context, we compare MATTRL with RareAgents augmented by few-shot exemplars (containing patient information and the final diagnosis). For each test case, 3 random exemplars are prepended to the conversation. As shown in Table 6, few-shot prompting yields only a minor improvement in Hit@1 while reducing Hit@3/5/10. This indicates that MATTRL's advantage arises from its structured experience integration rather than from simply adding more information.

Method	Hit@1	Hit@3	Hit@5	Hit@10
RareAgents	0.35	0.49	0.57	0.70
+ Fewshot	0.37	0.48	0.55	0.68
MATTRL	0.39	0.51	0.61	0.75

Table 6: Comparison with fewshot learning, where we add 3 examples at the beginning of each conversation.

6 Conclusion

We introduced **MATTRL**, a test-time adaptation framework that strengthens multi-agent reasoning by injecting *structured textual experience* into deliberation. MATTRL builds a small expert team, curates an experience pool from high-value dialogue turns via group-to-agent credit assignment, and retrieves these experiences to guide subsequent collaboration. Across **medicine**, **math**, and **education**, it consistently outperforms single- and multi-agent baselines, showing that experience-conditioned collaboration improves robustness under distribution shift. We further analyzed credit-assignment strategies and find that DIFFERENCE rewards provide a strong accuracy and efficiency trade-off for experience construction. Finally, an adaptive router that selects between single-agent inference and MATTRL yields additional gains by matching collaboration style to case complexity.

Limitations

We recognize two practical limitations remain. First, the method's inference-time compute and latency grow with multi-agent rollouts and exploration budget. Second, a continually growing test-time experience pool is vulnerable to drift: stale, duplicated, or spurious heuristics may accumulate. Looking ahead, we will (i) introduce dynamic budget controllers and confidence-based early stopping to cap cost without hurting accuracy, and (ii) add lifecycle management for experiences (recency weighting, de-duplication, anomaly screening) to preserve precision over time.

References

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, and 1 others. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6.

Xi Chen, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Hairui Li, Xue Zhang, Yingman Guo, Lei Fan, Gang Chen, and 1 others. 2025. Enhancing diagnostic capability with multi-agents conversational large language models. *NPJ digital medicine*, 8(1):159.

Xuanzhong Chen, Ye Jin, Xiaohao Mao, Lun Wang, Shuyang Zhang, and Ting Chen. 2024a. Rareagents: Autonomous multi-disciplinary team for rare disease diagnosis and treatment. *arXiv e-prints*, pages arXiv–2412.

Xuanzhong Chen, Xiaohao Mao, Qihan Guo, Lun Wang, Shuyang Zhang, and Ting Chen. 2024b. Rarebench: can llms serve as rare diseases specialists? In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4850–4861.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, and 1 others. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhitao He, Zijun Liu, Peng Li, Yi R Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Advancing language multi-agent learning with credit re-assignment for interactive environment generalization. In *Second Conference on Language Modeling*.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. 2025a. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*.

Zhiyuan Hu, Yibo Wang, Hanze Dong, Yuhui Xu, Amrita Saha, Caiming Xiong, Bryan Hooi, and Junnan Li. 2025b. Beyond'aha!': Toward systematic meta-abilities alignment in large reasoning models. *arXiv preprint arXiv:2505.10554*.

Yun Hua, Haosheng Chen, Shiqin Wang, Wenhao Li, Xiangfeng Wang, and Jun Luo. 2025. Shapley-coop: Credit assignment for emergent cooperation in self-interested llm agents. *arXiv preprint arXiv:2506.07388*.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Kartik Nagpal, Dayi Dong, Jean-Baptiste Bouvier, and Negar Mehr. 2025. Leveraging large language models for effective and explainable multi-agent credit assignment. *arXiv preprint arXiv:2502.16863*.

OpenAI. 2025. Introducing gpt-5.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. 2025. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2023. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and 1 others. 2024. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and 1 others. 2025. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*.

Jiayin Wang, Zhiquang Guo, Weizhi Ma, and Min Zhang. 2025. How far can llms improve from experience? measuring test-time learning ability in llms with human comparison. *arXiv preprint arXiv:2506.14448*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, and 1 others. 2025. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.

A Medicine

A.1 Detailed Setup

Task and data (RareBench Task 4). We instantiate MATTRL as an MDT-style workflow for rare-disease differential diagnosis on RareBench Task 4 (Chen et al., 2024b). Each instance provides a patient record $\mathcal{X}$ and the system outputs a ranked top-10 differential list. We evaluate with Hit@k and MRR as defined in the main text.

Agents, specialist pool, and recruitment. The system consists of a coordinator/chair agent LLM_{Coo} and a predefined specialist catalog $\mathcal{SP}$ (Appendix A.2). LLM_{Coo} recruits a small MDT $\text{TEAM} \subseteq \mathcal{SP}$ using the recruitment prompt (Appendix A.4), grounding role selection in real clinical departments rather than free-form role invention.

MDT interaction protocol and prompts. Given TEAM, specialists follow role-specific opinion prompts and produce a strict top-10 list each round (Appendix A.4). We run synchronized multi-round discussion with a maximum of $R_{\max}$ rounds as described in Sec. 3.1. The chair then synthesizes a discussion report and outputs the final ranked list using the final-decision prompt (Appendix A.4). Experience-augmented prompting uses the standardized injection template in Appendix A.3.

Utterance scoring and judge rubric. To score specialist utterances for experience construction, we use an LLM judge with the rubric defined in Appendix A.5, producing per-utterance scores $s_{i,t} \in [0, 1]$ (Eq. (9) in the main text). These individual scores are combined with a terminal case-level outcome signal via the decay-weighted allocation (Eq. (10)–(12) in the main text).

Experience extraction and summarization. High-scoring utterances are distilled into structured textual experiences using an LLM summarizer with the template in Appendix A.6. Each entry follows the `ACTION`/`EXPERIENCE` schema used in our experience-augmented prompt template (Appendix A.3).

Indexing and retrieval. At test time, each specialist retrieves relevant experience entries conditioned on the case and round context. We detail the embedding model, similarity metric, and top-K retrieval procedure in Appendix **??**. Retrieved experiences are appended to prompts via the injection block in Appendix A.3, keeping model weights fixed while providing dense guidance.

A.2 Description of Specialist Pool

This pool covers core inpatient and outpatient specialties frequently involved in complex differential diagnosis. It is designed to balance breadth with depth, enabling targeted and efficient MDT assembly.

Pediatrics	Urology
Hematology	Rheumatology
Psychiatry	Pulmonology
Dentistry	Endocrinology
Allergy and Immunology	Cardiology
Pathology	Neurology
Obstetrics and Gynecology	Ophthalmology
Dermatology	Geriatrics
Traditional Chinese Medicine	Nephrology
Oncology	General Practice
Gastroenterology	Infectious Diseases
Rehabilitation Medicine	Otorhinolaryngology

Table 7: List of 24 Departments from Specialist Pool.

A.3 Experience-Augmented Prompt Template

This template integrates retrieved experience into the base diagnostic instruction. The *Experience Context* block is formatted to remain model-friendly while improving calibration and coverage of edge patterns.

Experience-Augmented Diagnostic Prompt

You are an expert clinician AI agent participating in a diagnostic reasoning task.

Your goal is to reason step by step and propose the top 10 possible diagnoses for the given case.

Patient Case

`{question}`

Experience Context

Insert domain experience as structured hints; each item pairs an `ACTION` with an `EXPERIENCE`. The block is optional; omit if empty.

```
===== EXPERIENCE HINTS =====
- ACTION: <action_key_1>
  EXPERIENCE: <concise, actionable rule or
     prior finding>
- ACTION: <action_key_2>
  EXPERIENCE: <concise, actionable rule or
     prior finding>
...
===== END OF EXPERIENCE HINTS =====
```

Output Requirements
Return exactly the following structure; no extra prose outside it.

```
1) Reflection (2-3 sentences)
2) <diagnosis> block with exactly 10
    numbered items:
   1. [Diagnosis 1]: [1-2 sentence rationale
    ]
   ...
   10. [Diagnosis 10]: [1-2 sentence
    rationale]
</diagnosis>
```

Hard Constraints

- Base reasoning strictly on the Patient Case and the Experience Context (if provided).
- Do not invent external facts, tests, or treatments.
- Do not copy reasoning from other roles (if used in a multi-agent setting).
- If key information is missing, write "insufficient evidence" instead of guessing.

A.4 Prompts for Multi-disciplinary Team Collaboration

These prompts orchestrate role selection, role-specific reasoning, and peer oversight. The design favors minimal, structured outputs to simplify downstream aggregation and evaluation.

MDT Recruitment Prompt

You are the Chief Medical Officer assembling a single MDT.

Case: `{question}`

From this specialist pool: `{POOL}`

Pick no more than `N` distinct specialties best suited for this case. Never add unnecessary specialties just to complete the size.

Return **only** a valid JSON array. Each item must be:

```
{
  "specialty": "<one from the pool>",
  "role": "<short role name, exactly one
    item has 'leader'>",
  "description": "<2-4 sentences instructing
     how this specialist
  should reason for THIS case, including
    differential focus,
  red flags, and collaboration style>"
}
```

No prose outside JSON. No special characters.

Specialist Opinion Prompt

You are `{role}` in an MDT.
Goal: `{goal}`
Patient info: `{question}`

HARD CONSTRAINTS:

- Base reasoning strictly on Goal and Patient info.
- DO NOT introduce or infer external facts, tests, or treatments.
- DO NOT copy reasoning from other roles.
- If information is missing, write "insufficient evidence" instead of guessing.

OUTPUT FORMAT (STRICT):

```
1) Reflection (2-3 sentences)
2) <diagnosis> block with exactly 10
    numbered items:
   1. [Diagnosis 1]: [1-2 sentence rationale
    ]
   ...
   10. [Diagnosis 10]: [1-2 sentence
    rationale]
</diagnosis>
```

Peer Review Prompt

You are a `{reviewer role}` in a multidisciplinary team (MDT). Your task is to provide a critical peer review of the `{target role}`'s opinion on the current case.

Output strictly as JSON:

```
{
  "analysis": "3-4 sentences, concise expert
     analysis
              highlighting agreements and
    critical disagreements.",
  "agreements": [
    "Specific, constructive point of
    concurrence.",
    "Another key area of agreement with
    rationale."
  ],
  "disagreements": [
    "Precise point of contention with
    reasoning.",
    "Another targeted critique."
  ]
}
```

If you fully concur, set "disagreements" to `null`.

Final Decision Prompt

You are the chair of an MDT. Read the case snapshot and reason step by step.

MDT Investigations Summary: {assessment report}
Question: {question}

Output format:

```
<analysis>
- Summarize key findings and red flags.
- Explain differential logic and tie-
    breakers.
</analysis>

<top10>
[1] <Diagnosis name>
[2] <Diagnosis name>
...
[10] <Diagnosis name>
</top10>
```

Provide exactly 10 diagnoses. No extra text outside <analysis> and <top10>.

A.5 Rubrics for LLM Judge in Agent's Utterance

This rubric converts free-form predictions into a single categorical judgment for evaluation. The instructions prefer clinical synonymy while rejecting incompatible subtypes, balancing sensitivity and specificity for leaderboard scoring.

Evaluation Prompt

I will now give you ten predicted diseases. If the predicted diagnosis is in the standard diagnosis list, output its rank (1–10); otherwise, output "No". Output exactly one value—either "No" or a single number from 1 to 10. If multiple match, choose the highest rank.

Decide whether the predicted disease and a standard diagnosis are the **same medical condition**. Be moderately strict: allow synonyms and parent unspecified subtype matches, but do not accept clearly distinct subtypes or genetic forms as the same.

Matching Rules:

- **ACCEPT** if they are synonyms, eponyms, abbreviations, or different wording for the same condition.
- **ACCEPT** if one is a broad parent category and the other is an unspecified form within that category.
- **REJECT** if they specify different subtypes (e.g., type I vs type II), different enzyme defects, or different genes.
- **REJECT** if they are unrelated conditions or only partially overlapping.

Output Format:

```
<think>
Step-by-step reasoning:
</think>
<answer>No|1-10</answer>
```

Example Input:
Predicted diseases: {predict_diagnosis}
Standard diagnosis: {golden_diagnosis}

A.6 Prompts for LLM Summarizer

The summarizer condenses multi-turn MDT content into an actionable brief for clinicians or downstream modules, emphasizing signal over verbosity and avoiding speculative language.

Prompts for MDT Chair Summarizer

You are the chair of an MDT. Read the case snapshot and reason step by step.
MDT Investigations Summary: {assessment_report}
Question: {question}
Output format (STRICT): <analysis> - Summarize key findings and red flags. - Differential logic: why certain classes of diseases are prioritized. - Tie-breakers and evidence weighting. </analysis>
<top10> [1] <Diagnosis name>[2] <Diagnosis name>[3] <Diagnosis name>[4] <Diagnosis name>[5] <Diagnosis name>[6] <Diagnosis name>[7] <Diagnosis name>[8] <Diagnosis name>[9] <Diagnosis name>[10] <Diagnosis name></top10>
Rules:
- Provide exactly 10 diagnoses, one per line, each starting with [rank].
- Be precise and avoid variants on the same concept unless clinically distinct.
- No extra text outside <analysis> and <top10>.

A.7 Retrieval Implementation Details

We implement the retrieval module $\mathcal{M}$ using a dense vector index to inject relevant reasoning priors. Specifically, we employ `Qwen/Qwen3-Embedding-4B` as the backbone encoder $E(\cdot)$. To ensure the inner product search is equivalent to cosine similarity, we apply L_2 normalization to the embeddings of all key-value experience pairs (k_i, v_i) stored in the database, yielding index vectors $\mathbf{u}_i = E(k_i)/\|E(k_i)\|_2$, which are stored using the `FAISS` library's `IndexFlatIP`. During inference at time t, the current agent's instruction x_t is encoded into a normalized query vec-

tor $\mathbf{q}_t = E(x_t)/\|E(x_t)\|_2$. The system retrieves the top-K entries (default $K = 8$) by maximizing the similarity score $s_i = \mathbf{q}_t^\top \mathbf{u}_i$ and appends them to the prompt using a strict "`EXPERIENCE HINTS`" template to guide the model's reasoning.

B Mathematics

B.1 Detailed Setup

We instantiate MATTRL for multi-agent mathematical problem solving (Figure 4). Given a math problem (task record $\mathcal{X}$), the coordinator agent $\mathrm{LLM}_{\mathrm{Coo}}$ forms a small team of specialists, runs a bounded multi-round collaboration with optional experience retrieval, and finally synthesizes a discussion report and outputs the final solution. This appendix specifies the concrete collaboration protocol and prompts used in the math setting.

Baseline run (no experience) We first run MATTRL (math) with experience augmentation disabled (`-use_experience` off). For each problem, the pipeline outputs a final solution artifact and a detailed interaction log recording each specialist opinion, peer review, and round summary.

B.2 Free recruitment team formation

In math, because of the flexibility of math problem, we use free recruitment: instead of selecting from a fixed catalog, the coordinator directly proposes a small set of specialist descriptions tailored to the current problem, and forms TEAM accordingly. This corresponds to the team-formation stage of our pipeline, where the coordinator constructs a small set of role-specialized agents on-the-fly for each problem.

Recruitment Prompt for Free Recruitment

You are the Chief Problem Solving Officer assembling a small math team.

Problem: `{problem}`

Design up to `{n}` distinct specialties best suited for this problem.
You are NOT restricted to any preset pool. Create precise specialist titles that reflect reasoning roles (e.g., invariant designer, structure normalizer, edge-case auditor).
Exactly ONE item must have role `leader`.

Return ONLY a valid JSON array. Each item must be:

```
{
  "specialty": "<your created specialist
    title>",
  "role": "<short role name, exactly one
    item has 'leader'>",
  "description": "<2-4 sentences on how this
     specialist will reason for THIS problem
    :
                  key invariants, typical
    tactics, how to communicate>"
}
```

No prose outside JSON.

B.3 Multi-round collaboration (Stage II)

We run up to $R_{\max}$ collaboration rounds. In each round, every non-converged specialist proposes a solution attempt; other specialists then provide targeted critiques and minimal fixes in a structured format. The coordinator aggregates these critiques into a concise feedback bulletin, which is provided to specialists in the next round for revision. A specialist is marked converged once their solution no longer changes under critique.

Round 1 Specialist Opinion Prompt

You are `{specialty} ({role})` in a math problem-solving team.
Goal: `{team_goal}`
Problem: `{problem}`

Please think step by step from your expert perspective, and produce ONE integrated, concise solution message.
Include: brief reflection (1–2 sentences), core reasoning leading to the result, and the final answer (one line if known).
Keep it rigorous and self-contained; avoid decorative tags.

B.4 Structured peer review and acceptance rule

For each specialist's attempt, all other specialists generate a structured peer review in raw JSON, including an overall appraisal, a verdict (`accept/revise/reject`), validated parts, and a list of concrete issues with severities (`fatal/major/minor`) and minimal fixes. A specialist's attempt is marked accepted *only if* (i) all peer verdicts are `accept` and (ii) the `issues` list is empty. When critiques identify no remaining issues, we treat the specialist's update as converged (i.e., no further changes are proposed in subsequent rounds). The collaboration halts when all specialists converge or when reaching round budget.

Structured Peer Review Prompt

You are reviewing another specialist's solution attempt from your own perspective.

Context bundle (JSON):
`{bundle_json}`

DISCIPLINE:
- Keep `analysis` to 3–4 concise sentences: start with grounded agreements, then targeted critiques.
- Put parts you believe correct in `validated` as short phrases.
- For EACH critique, add an entry in `issues` with type, severity, concrete note, and a minimal fix.
- Use `tests` to note quick sanity checks (toy examples, boundary cases, counterexamples).

Verdict rule: `accept` only if there are NO issues at any severity (including minor) and the solution is essentially complete;
`revise` if ANY fixable issue remains; `reject` if there is a fatal flaw.

OUTPUT (RAW JSON ONLY):

```
{
  "analysis": "2-4 sentences overall
    appraisal.",
  "verdict": "accept | revise | reject",
  "validated": ["..."],
  "issues": [
    {"type":"algebra_error|logic_gap|
    missing_case|theorem_misuse|
    undefined_symbol|diagram_dependence...",
     "severity":"fatal|major|minor",
     "note":"Exactly what is wrong.",
     "fix":"Concrete minimal correction
    consistent with the current approach."}
  ],
  "tests": {"random_checks": 0, "
    counterexample": {"found": false, "
    example": null}},
  "alt_hint": "Optional short hint to guide
    a revision.",
  "confidence": 0
}
```

Revision Opinion Prompt with Peer Feedback

You are `{specialty}` (`{role}`) in a math team.
Problem: `{problem}`

Last accepted snapshot (may be empty):
`{accepted_snapshot}`

Peer feedback for YOU (JSON):
`{delta_json}`

Please think step by step from your expert perspective, and produce ONE integrated, concise message addressing feedback (no step numbering).
State the refined reasoning and the final answer if applicable.
Be precise and minimal; no special tags.

B.5 Chair aggregation (final decision)

After bounded discussion, the coordinator $\mathrm{LLM}_{\mathrm{Coo}}$ synthesizes a discussion report DR from all specialists' updates (Stage III), and outputs the final solution. If the first chair output does not contain these tags, the system triggers a rewrite pass that preserves mathematical content but enforces the target format.

Chair Aggregation Prompt

You are the chair of a math MDT. Integrate accepted outlines and produce a concise, rigorous write-up.

Snapshot summary (accepted/unresolved):
`{assessment_report}`

Problem:
`{problem}`

OUTPUT FORMAT (STRICT):

```
<analysis>
- Key ideas and why this route is effective.
- Note any edge cases handled.
</analysis>

<final_answer>
- Single line with the final numeric/
    symbolic answer; if unknown, write 'N/A
    '.
</final_answer>

<formal_proof>
Provide a clean, step-by-step solution/proof.

</formal_proof>
```

No extra text outside these tags.

B.6 Rubrics for LLM Judge in Math Utterance Scoring

We use an LLM judge to score (i) the terminal correctness of the final answer and (ii) the per-utterance contribution within the multi-agent transcript. The terminal judgment provides the team outcome signal $G \in \{0, 1\}$, while the utterance-

level score $s_{i,t}$ measures how much a given agent utterance helps (or hurts) reaching the correct final solution. In implementation, the judge outputs an integer score in $[0, 5]$, and we optionally normalize it to $[0, 1]$ by $s_{i,t} = \text{score}/5$.

Math Judge Prompt: Terminal Answer Correctness

You are a strict math judge. Determine whether the predicted final answer matches the golden answer (allowing standard mathematical equivalence).

Problem: `{problem}`
Predicted Final Answer: `{final_answer}`
Golden Answer: `{golden}`

Output format (STRICT):

```
<analysis>
Briefly justify equivalence/non-equivalence.
</analysis>
<answer>
Yes|No
</answer>
```

Math Judge Prompt: Utterance Influence Scoring (0–5)

You are a math reasoning evaluator. Your task is to score how much the target agent utterance influences reaching the correct final answer.
Score range: 0 (no helpful influence) to 5 (decisive helpful influence).

Context you will be given:
- Problem statement
- Step 0 snapshot (baseline)
- Step i snapshot (with the target utterance)
- Predicted final answer and golden answer
- The target agent's prompt and response

Output STRICT JSON only:

```
{
  "analysis": "1-3 sentences explaining why
    this utterance helps/hurts.",
  "score": 0-5
}
```

We then combine the utterance score with the decayed terminal signal: the terminal correctness G is allocated to turns with a decay factor and distributed among agents within the same turn proportionally to their utterance scores, and finally fused with the direct utterance score to obtain $r_{i,t}$ used for experience selection.

B.7 Interaction scoring and selection (train split only)

We score each specialist utterance with an LLM judge to obtain an individual score $s_{i,t}$, then combine it with a terminal correctness signal allocated back to turns using a decay factor. Each specialist utterance is scored by an LLM judge to obtain an individual score $s_{i,t}$, and a terminal correctness signal G is allocated back to turns with decay. We then select high-value utterances (e.g., top quantile or thresholded by $r_{i,t}$) to form the candidate set for experience extraction.

B.8 Experience extraction and indexing (train split only)

Selected high-value utterances are distilled into concise textual experiences using a fixed LLM-based summarization template, producing key–value entries that are easy to retrieve. We embed the keys, build a dense index (Appendix A.7), and retrieve top-K experiences at inference time. Retrieved experiences are appended to prompts using the standardized `EXPERIENCE HINTS` block.

B.9 Test-Time Experience Retrieval

At test time, each non-converged specialist retrieves relevant experiences from the shared pool $\mathcal{E}$ based on the current problem and its round context. Retrieval is implemented with dense embeddings and a FAISS index. The retrieved entries are appended to the prompt using the same `EXPERIENCE HINTS` template as other domains, serving as consultable guidance without updating model weights.

Test-time Experience Hint Injection

When experience augmentation is enabled, retrieved hints are appended to the end of the prompt:

```
===== EXPERIENCE HINTS (consult; do not
    quote verbatim) =====
- ACTION: <retrieved key 1>
  EXPERIENCE: <retrieved experience 1>
- ACTION: <retrieved key 2>
  EXPERIENCE: <retrieved experience 2>
...
===== END OF EXPERIENCE HINTS =====
```

Agents may consult these hints to improve reliability (e.g., invariant choice, missing cases, counterexample checks), but should not quote them verbatim in the final solution.

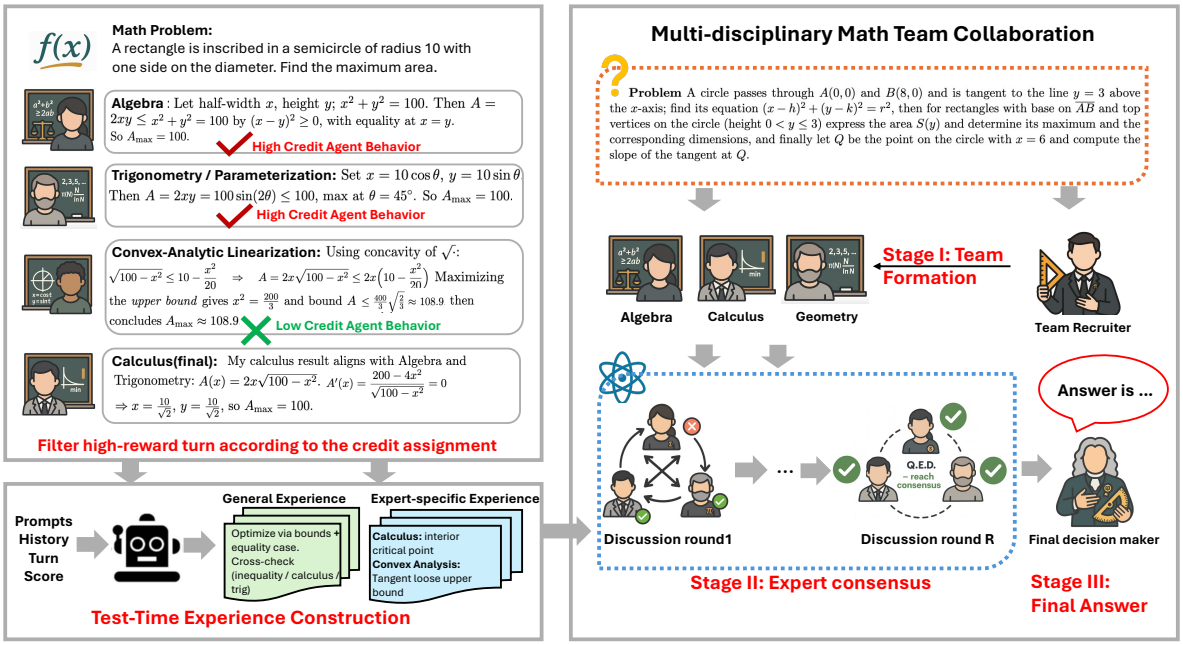

Figure 4: MATTRL in Math: Multi-Specialist Math Problem-solving Collaboration.

C Education

C.1 Detailed Setup

Large language models are increasingly serving as educational tools, yet evaluating their teaching capabilities remains challenging. In this experiment, we adapt the MATTRL framework and create a realistic learning scenario in which a team of pedagogy specialists works together to guide students through complex problem-solving tasks (Figure 5). This setup allows us to test how effective the MARRLL is at improving the teaching performance of multi-agent systems.

Pre-test A pre-test is conducted to establish baseline student performance before any instruction. A student agent (GPT-4o, temperature=0.3) is prompted (in C.4.1) to solve multiple-choice questions from SuperGPQA, providing both the answer and reasoning to surface its thinking and uncertainties. Pre-test questions are selected via stratified sampling across 13 subject matters and three difficulty levels to ensure balanced coverage. The pre-test is run once before any teaching sessions, and the same student agent instance is reused across all experimental conditions.

Pedagogy Specialist Team Formation Before the instructional session, a pedagogy specialist team of three members is formed based on an analysis of the question and the students' pre-test performance. Team members are selected from a predefined pool (C.4.2, Table C.3) that includes subject-matter experts, pedagogical specialists, and cross-disciplinary specialists. Each team member is assigned a specific role: the diagnostician identifies the reasons for the student's incorrect response, the pedagogy strategist proposes appropriate instructional strategies, and the subject matter expert provides discipline-specific explanations.

Multi-round teaching session During the teaching session, the teaching agent (GPT-5, temperature=0.3) is provided with the full question text and the correct answer, the student's pre-test response and reasoning, and the correctness status. The teacher agent guides the student toward the correct answer through a structured, three-round question–answer dialog that diagnoses and clarifies misconceptions while scaffolding the student's reasoning, without directly revealing the answer. Three teaching conditions are evaluated for comparison: (1) a *Single-Teacher* condition, in which a single agent conducts the full dialog using a fixed instructional prompt (C.4.3); (2) a *Multi-Teacher* condition, in which multiple specialist agents generate each instructional strategy analysis based on their role-specific perspectives first and collaboratively plan before interacting with the student agent (Prompt: C.4.4); and (3) a *Multi-Teacher with Experience* condition, which extends the collabora-

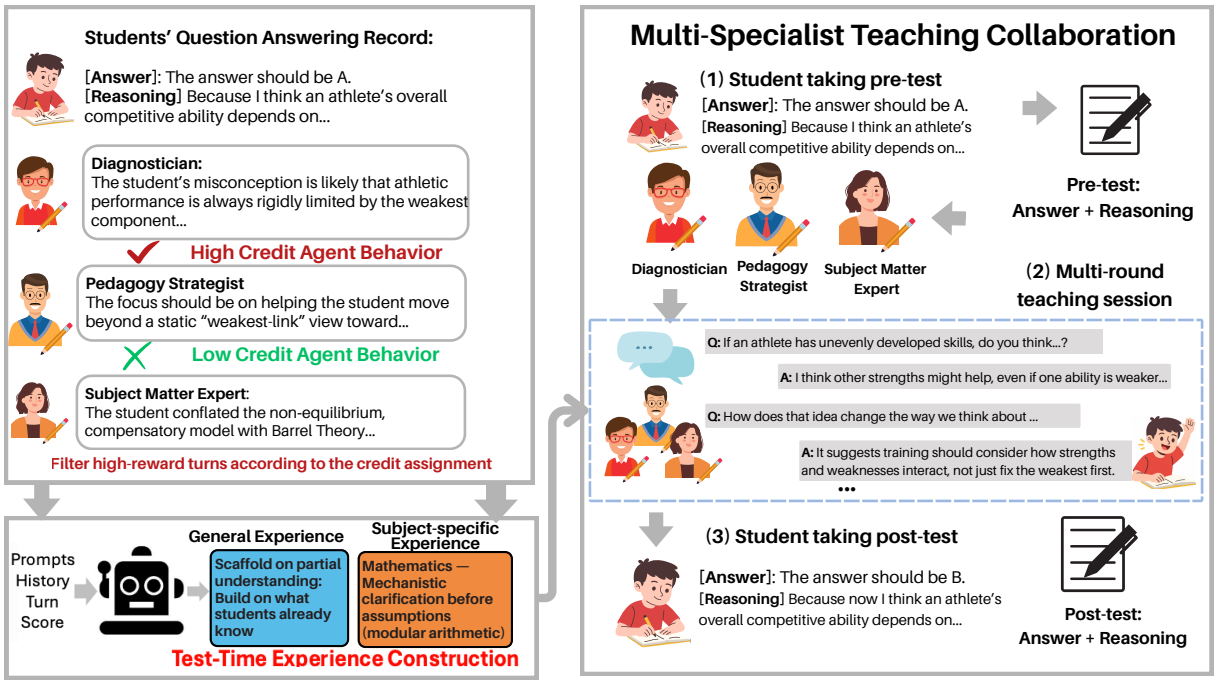

Figure 5: MATTRL in Education: Multi-Specialist Teaching Collaboration.

tive setting by incorporating role-, subject-, and difficulty-specific teaching experiences retrieved from the experience pool to inform instructional strategy generation (Prompt: C.4.5).

Post-test In the post-test, the student agent answers the same question again using the same response format as the pre-test. If the student answered correctly on the pre-test, the teaching session will be skipped, and the pre-test answer will be reused in the post-test.

Interaction scoring and selection To construct the pedagogy experience pool, additional teaching interactions are generated using stratified sampling over subject domains and difficulty levels from the SuperGPQA dataset under the multi-agent teaching setting described above. 28 successful cases are finally identified and scored using two complementary signals. First, a global outcome score captures overall instructional success and is defined as a binary indicator of post-test correctness, assigning a value of 1.0 if the student's post-test answer is correct and 0.0 otherwise. Second, a step-level influence score evaluates the contribution of each teaching-strategy utterance to student learning. Each utterance is rated on a 0–5 scale by an LLM adjudicator, measuring its causal influence on the student's progress relative to the pre-test baseline. In addition, each role of teachers' pedagogy analyzing utterance is evaluated using a rubric-based utterance quality score (C.4.6). The binary global outcome score is temporally allocated across dialogue turns using decay with factor $\gamma = 0.85$, assigning higher credit to earlier instructional turns. Within each turn, the allocated global credit is distributed across utterances in proportion to their step-level influence scores. Finally, each utterance is assigned a combined score computed as a weighted average of its share of the decayed global credit and its direct instructional contribution, with weights of 0.6 and 0.4, respectively.

Experience extraction and summarization From each scored teaching interaction, the top-ranked (25%) utterances are selected based on their *final_score* and converted into transferable pedagogical experiences using an LLM-based extractor (C.4.7). Each extracted experience follows a constrained instructional format and is categorized as either general or subject-specific. Experiences are indexed by the teacher role, subject domain, and difficulty level, and stored in a structured format. We provide example experiences here in C.2.

C.1.1 Test-Time Experience Retrieval

At test time, when experience augmentation is enabled, each teacher agent first attempts to load a role-specific pedagogy experience knowledge base according to the corresponding question subject matter and difficulty level. The role-specific knowledge base is identified using the agent's assigned

instructional role (e.g., Diagnostician, Subject Matter Expert, or Pedagogy Strategist). Retrieved experiences are appended to the agent's prompt in a *Experience Hints* section, explicitly marked as consultative guidance intended to inform the agent's instructional decisions, rather than to be quoted verbatim in generated responses.

C.2 Experience Examples

Experiences

General Teaching Experience:

Scaffold on partial understanding: Identify and leverage what students already understand, using existing knowledge as a foundation for introducing new concepts.

Anchor on key discriminators first: Organize instruction around the core distinctions or decision points that differentiate correct from incorrect reasoning, using them as the backbone of explanation.

Explicitly name misconceptions: Surface and label specific misconceptions directly, rather than correcting them implicitly, to prevent persistent or repeated errors.

Subject-Specific Teaching Experience:

Mathematics: Prioritize mechanistic clarification before introducing assumptions, particularly in structured domains such as modular arithmetic, where procedural misunderstandings often precede conceptual errors.

Philosophy: Emphasize precise definitions and logical scope, especially in tasks involving entailment, quantifiers, or closely related concepts, where minor definitional shifts can alter reasoning outcomes.

Medicine: Apply a high evidentiary standard when distinguishing descriptive from causal claims, with particular attention to study design and classification to avoid overgeneralization or misinterpretation.

Figure 6: General & subject-specific experience

C.3 Description of Specialist Pool

This specialist pool spans key academic domains, pedagogical expertise, and cross-disciplinary perspectives, enabling flexible and targeted formation of specialist teams for instructional support.

Category	Specialist Pool
Domain Experts	Mathematics; Engineering; Physics; Chemistry; Biology; Computer Science; Medicine; Agriculture; Economics; Management; Law; Education; Military Science; History; Literature; Philosophy; Sociology; Language Arts
Pedagogical Specialists	Pedagogy; Educational Psychology; Assessment and Evaluation; Curriculum Design
Cross-Disciplinary Specialists	STEM; Humanities; Social Sciences

Table 8: Specialist pool used for pedagogy team formation.

C.4 Prompts

C.4.1 Prompt for Student Agent in Pre-test

This prompt guides the student agent to answer a multiple-choice question while explicitly articulating its reasoning process.

Prompt for Student Agent in Pre-test

Answer the following multiple-choice question. There is only one correct answer.
Before giving the final answer, briefly explain: (1) Your reasoning for how the correct option is determined (2) Any uncertainties, ambiguities, or alternative interpretations you considered.
The last line of your response should be in the format: Answer: $LETTER' (without quotes),
(where LETTER is one of A, B, C, D, E, F, G, H, I, or J.
[Question text]
A) [Option A]
B) [Option B]
...

C.4.2 Pedagogy Specialist Recruitment Prompt

This prompt guides the pedagogy specialist to assemble an appropriate teaching team by identifying the pedagogical expertise required for the given pre-test question.

Pedagogy Specialist Recruitment Prompt

You are an experienced educational coordinator assembling a teaching team.

Pre-test Question: `{question}`

IMPORTANT: Your team MUST include these key pedagogical roles: 1. The Diagnostician: Analyzes why the student failed the pre-test (e.g., calculation error vs. conceptual misunderstanding) 2. The Pedagogy Strategist: Proposes the next move (e.g., 'Ask a probing question' vs. 'Provide a counter-example') 3. Subject Matter Experts: Provide disciplinary knowledge explanation (select 1-2 from the pool based on question topic)

From this teacher/specialist pool: `{POOL}`

Pick no more than n distinct specialties best suited for this question. Assign roles so that you have:
- At least one Diagnostician role (can assign to any specialty, but they focus on analyzing student errors)
- At least one Pedagogy Strategist role (can assign to any specialty, but they focus on teaching strategy)
- 1-2 Subject Matter Experts (actual domain specialists from the pool)
- Optionally one lead teacher to coordinate

Never add unnecessary specialties just to complete the size.
Return **only** a valid JSON array. Each item must be:

```
{
  "specialty": "<one from the pool, or use a
     pool specialty name>",
  "role": "<one of: Diagnostician, Pedagogy
    Strategist, Subject Matter Expert, lead
    teacher>",
  "description": "<2-4 sentences instructing
     an LLM acting as this teacher on how to
     approach THIS question.
                 If role is Diagnostician,
    focus on error analysis. If role is
    Pedagogy Strategist, focus on teaching
    methods.
                 If role is Subject Matter
    Expert, focus on domain knowledge.
    Include how to collaborate with other
    teachers>"
}
```

No prose outside JSON. No special characters.

C.4.3 Prompts for Single-Teacher Instruction

This prompt guides the teacher agent to generate instructional feedback based on the student's pre-test answers and reasoning.

Prompt for Single-Teacher Instruction

Initialization:
Question: `{question}`
Student's Pre-Test Answer: `{answer}`
Student's Reasoning: `{reasoning}`
Correct Answer: `{gold answer}`
Your task: Guide the student to arrive at the correct answer through a `{num rounds}`-round dialogue.
- You know the correct answer is `{gold answer}`, but DO NOT directly reveal it
- Ask thoughtful questions to guide their thinking toward the correct answer
- Address any misconceptions in their reasoning
- Help them discover the correct answer through guided discovery
- Focus on teaching concepts and reasoning that lead to the correct answer

Round 1 Prompt:
You are teaching a student who just took a pre-test.
Here's what they did:
Question:`{question}`
Student's Answer: `{answer}`
Student's Reasoning: `{reasoning}`
Based on the student's answer and reasoning, ask them a thoughtful question to guide their thinking. Ask your question now (just the question, no preamble):

Student Response Prompt:
The teacher just asked you: `{question}` Please respond to the teacher's question thoughtfully.

Round 2 Prompt:
The student just responded to your previous question. Based on their response, ask a follow-up question to continue guiding them. Ask your follow-up question now (just the question, no preamble):

Final Guidance Prompt:
Now that you've had a dialogue with the student, provide final teaching guidance:
- Summarize key concepts they should understand
- Clarify any remaining misconceptions
- Guide them on how to approach this type of problem correctly
- Explain the underlying principles and reasoning process
- DO NOT directly state which option/letter is the correct answer

C.4.4 Prompt for Multi-Teacher Instruction

This prompt guides the teacher agent to generate instructional feedback based on the student's pre-test answer and reasoning.

Prompt for Multi-Teacher Instruction

Round 1 - Individual Analysis Prompt
You are `{role}`, a specialized teacher analyzing a student's pre-test response.
FULL QUESTION:`{question}`
STUDENT'S PRE-TEST RESPONSE:
- Answer: `{answer}`
- Reasoning: `{reasoning}`
- Correct: `{correct}`

CORRECT ANSWER: `{gold answer}` (You know this, but DO NOT reveal it directly to the student)
YOUR TASK (based on your role):`{role instructions}` Analyze the teaching strategy from your perspective. Provide your analysis in 2-3 sentences.
ROLE-SPECIFIC INSTRUCTIONS:
Diagnostician: focus on identifying WHY the student failed
Pedagogy Strategist: focus on HOW to teach this student
Subject Matter Expert: focus on WHAT domain knowledge the student is missing or misunderstanding

Round 1 - Collaborative Planning Prompt:
You are a team of specialized teachers collaborating to plan a 2-round teaching dialogue.
FULL QUESTION:`{question}`
STUDENT'S PRE-TEST RESPONSE:
- Answer: `{answer}`

- Reasoning: {reasoning}
CORRECT ANSWER: {gold answer (You know this, but DO NOT reveal it directly)
INDIVIDUAL TEACHER ANALYSES:{analyses summary}
YOUR COLLABORATIVE TASK:
Based on all the analyses above, work together to plan specific and targeted questions that will guide the student to discover the correct answer.
Requirements:
1. Each question should be specific and directly related to the student's reasoning
2. DO NOT directly state which option/letter is correct
Return your planned questions in this EXACT format:
ROUND 1: [specific question here]
Student Response Prompt:
The teacher just asked you: {question} Please respond to the teacher's question thoughtfully.
Round 2 - Individual Analysis Prompt
The student just responded to Round {num}:
Teacher's Question: {question}
Student's Response: {response}
Based on your role ({teacher.role}), analyze the teaching strategy from your perspective. Provide your analysis in 2-3 sentences.
Round 2 - Collaborative Planning Prompt:
You are a team of specialized teachers collaborating to generate the next question for Round {num + 1.
CONVERSATION HISTORY SO FAR: {history}
STUDENT'S LATEST RESPONSE (Round {num}):{student response}
INDIVIDUAL TEACHER ANALYSES:{analyses summary}
YOUR COLLABORATIVE TASK:
Based on all the analyses above, work together to plan a specific and targeted question for Round {num + 1} that will guide the student to discover the correct answer
Return your planned questions in this EXACT format:
ROUND 2: [specific question here]
Final Guidance Prompt:
Now that you've had a dialogue with the student, provide final teaching guidance:
- Summarize key concepts they should understand
- Clarify any remaining misconceptions
- Guide them on how to approach this type of problem correctly
- Explain the underlying principles and reasoning process
- DO NOT directly state which option/letter is the correct answer

C.4.5 Prompt for Experience Integration

This prompt guides the teacher agent to incorporate retrieved teaching experiences into instruction.

Prompt for Experience Integration

When experience augmentation is enabled, retrieved hints are appended to the end of the prompt:

```
===== USING TEACHING EXPERIENCES =====
Below are teaching experiences from similar
    successful cases. Use them to INFORM
    your strategic thinking, but ADAPT them
    to THIS specific student's error.

HOW TO USE EXPERIENCES:
1. Look for patterns in WHY students made
    similar errors - does this student's
    error match?
2. Identify WHERE similar problems occurred -
     does this help you locate the current
    student's error?
3. Learn HOW successful teachers addressed
    similar misconceptions - what strategies
     worked?
4. Adapt the strategies to THIS student's
    specific reasoning and error pattern
5. DO NOT blindly copy - each student's
    error is unique, even if similar

===== EXPERIENCE HINTS (for the role to
    consult; do not quote verbatim) =====
{experience_hints}

===== END OF EXPERIENCE HINTS =====

REMEMBER: Experiences are GUIDANCE, not
    rules. Your primary focus is THIS
    student's specific error, reasoning, and
     path to the correct answer ({
    gold_answer}).
```

C.4.6 Prompt for Teaching Utterance Evaluation

This prompt guides an expert evaluator agent to assess a teaching utterance across multiple instructional quality dimensions, including correctness, information gain, relevance, and clarity.

Prompt for Teaching Utterance Evaluation

You are an expert educational evaluator.
Evaluate a teacher's utterance based on these criteria:
1. CORRECTNESS (30%): Is the utterance factually correct and accurate?
2. INFORMATION GAIN (25%): Does it provide useful information for student to learn and get closer to

the correct answer?
3. RELEVANCE (25%): Is it relevant to the student's misconception or the learning task?
4. CLARITY (20%): Is it clear, understandable, and well-structured?
Question: {question}
Student's Pre-Test Answer: {pretest answer}
Student's Pre-Test Reasoning: {pretest reasoning}
Correct Answer: {gold answer}
Teacher Utterance:{teacher utterance}
Output STRICT JSON:

```
{
  "correctness_score": 0-5,
  "information_gain_score": 0-5,
  "relevance_score": 0-5,
  "clarity_score": 0-5,
  "analysis": "<brief explanation of scores
    >",
  "global_score": 0-5
}

Calculate global_score as weighted sum:
global_score = 0.30 * correctness_score +
    0.25 * information_gain_score + 0.25 *
    relevance_score + 0.20 * clarity_score
Round to nearest integer (0-5).
```

When experience augmentation is enabled, retrieved hints are appended to the end of the prompt:

C.4.7 Prompts for Experience Summarizer

This prompt guides an experience summarizer agent to extract and structure reusable teaching guidance and strategies from teaching interactions.

Prompts for Experience Summarizer

You are an expert educational experience analyzer.
TASK: From a ROLE, PROMPT, RESPONSE, GROUND TRUTH, EVAL ANALYSIS, FINAL SCORE pair, produce retrieval-ready key:value experiences in TWO classes:
1) GUIDANCE — how a teacher handles a teaching situation or student error.
2) STRATEGY — how a teacher implements a specific teaching approach.
INPUTS
- PROMPT ROLE: prompt role
- INPUT PROMPT: source text containing instructions, constraints, student's answer/reasoning, question context.
- INPUT RESPONSE: produced teaching utterance/question/guidance.
- GROUND TRUTH (optional): Correct answer
- EVAL ANALYSIS (optional): Evaluation analysis of the utterance's effectiveness
- FINAL SCORE (optional numeric): Final score of the utterance
OUTPUT
Return ONE JSON object ONLY where each entry is: "KEY": "EXPERIENCE".
KEY CONSTRUCTION (single line, natural English, no brackets):
- Format: "CLASS — Role action type teaching situation in Subject Difficulty: concise content"
- CLASS is either "GUIDANCE" or "STRATEGY"
- Role is the teacher's role (e.g., Diagnostician, Pedagogy Strategist, Subject Matter Expert)
- teaching situation describes the teaching context (e.g., "misconception", "error analysis", "question generation")
- Subject and Difficulty come from the question metadata
EXPERIENCE (value) — 1-2 sentences, must:
- Be generalizable, instructional teaching guidance (how to identify errors, how to guide students, how to address misconceptions, how to structure effective questions).
- Include a clear judgment prefix:
* "Good practice: ..." when behavior aligns with successful teaching outcomes (student corrected their answer, showed understanding).
* "Pitfall: ..." when behavior conflicts with teaching effectiveness or adds confusion.
- Optionally append a simple outcome tag at the end: [helpful] / [harmful] / [neutral] / [insufficient].
- Justify judgment ONLY by contrasts observable between INPUT RESPONSE and GROUND TRUTH / EVAL ANALYSIS / FINAL SCORE. If these are absent or inconclusive, use "Pitfall/Good practice" only if you can justify from INPUT RESPONSE behavior; otherwise end with [insufficient].