

Bias-Aware Orchestration: Role-Specialized Multi-Agent Collaboration of Heterogeneous LLMs

Yuanpei Zhao¹, Haowei Zhang¹, Ziyue Qiao²,
Jiancheng Lv¹, Mao Li^{1,†}, Wei Ju^{3,†}

¹College of Computer Science, Sichuan University

²School of Computing and Information Technology, Great Bay University

³School of Artificial Intelligence, Sichuan University

zhaoyuanpei@stu.scu.edu.cn, zhanghaowei1@stu.scu.edu.cn

ziyuejoe@gmail.com, lvjiancheng@scu.edu.cn

limao@scu.edu.cn, juwei@scu.edu.cn [†]Corresponding authors.

Abstract

Coordinating heterogeneous large language models (LLMs) can improve instruction following, yet current multi-agent systems often depend on manual role choices or repeated calls to one backbone. This makes it difficult to separate complementary model behavior from model bias and verbosity. We introduce Role-Specialized Multi-Agent Collaboration (RS-MA), a prompting-only framework that assigns candidate LLMs to four interpretable roles: Judge, Summarizer, Ideator, and Reasoner. RS-MA estimates role fit with audition prompts, aggregates role-specific rankings with Borda scoring, and uses the selected assignment as an inspectable role map. A refinement phase records judgment gaps and targeted edits before synthesis. On AlpacaEval 2.0, audition-selected RS-MA reaches an 88.86% length-controlled win rate, and a validated role map reaches 90.72%. A full 805-example ablation shows that single-model reuse attains high raw preference through much longer outputs, whereas the validated map improves length-controlled quality with moderate length growth. MT-Bench and FLASK further indicate preserved dialogue quality and balanced fine-grained skills. The results support explicit role assignment as a practical mechanism for combining heterogeneous LLMs when quality, completeness, and inspectable evidence are jointly required.

1 Introduction

Large language models (LLMs) have advanced rapidly in instruction following, reasoning, and multimodal understanding. Many real tasks, however, require several capabilities at once, such as creative proposal generation, factual checking, boundary analysis, and coherent synthesis. A single model instance often exposes one dominant bias profile, while multi-agent systems can distribute these functions across specialized agents (Wu et al.,

2023; Li et al., 2023; Liang et al., 2024; Xie et al., 2023; Chen et al., 2024b).

Recent orchestration methods show the value of structured collaboration. Role-playing systems such as CAMEL encourage division of labor (Li et al., 2023). Debate and consensus methods elicit competing views and aggregate them through adjudication (Liang et al., 2024). General platforms such as AutoGen, AgentVerse, and OpenAgents provide reusable conversation loops, planning modules, and tool interfaces (Wu et al., 2023; Xie et al., 2023; Chen et al., 2024b; Wang et al., 2025d). Mixture-of-Agents (MoA) further shows that combining outputs from multiple LLMs can improve average answer quality (Wang et al., 2025b). These systems establish multi-agent collaboration as a useful inference-time strategy.

Despite this progress, three practical issues remain. First, many pipelines use the same backbone for all agents, which couples failure modes and limits cross-model complementarity. Second, aggregation is often implicit or sensitive to response length, making it difficult to know whether a judged improvement comes from better content or from a longer answer (Dubois et al., 2024). Third, most systems do not formalize which model should play which role. Roles are usually defined by prompts, while the mapping between model bias and role demand remains ad hoc (Guo et al., 2024; Tao et al., 2024; Dong et al., 2024; Nguyen et al., 2025). In this paper, *bias-aware* refers specifically to model-role mismatch in orchestration and length bias in preference evaluation by model judges.

We propose Role-Specialized Multi-Agent (RS-MA), a prompting-only framework for heterogeneous LLM collaboration. RS-MA organizes models into four interpretable roles: Judge, Summarizer, Ideator, and Reasoner. The system runs a two-phase refinement loop with candidate generation, gap-aware judging, structured synthesis, targeted

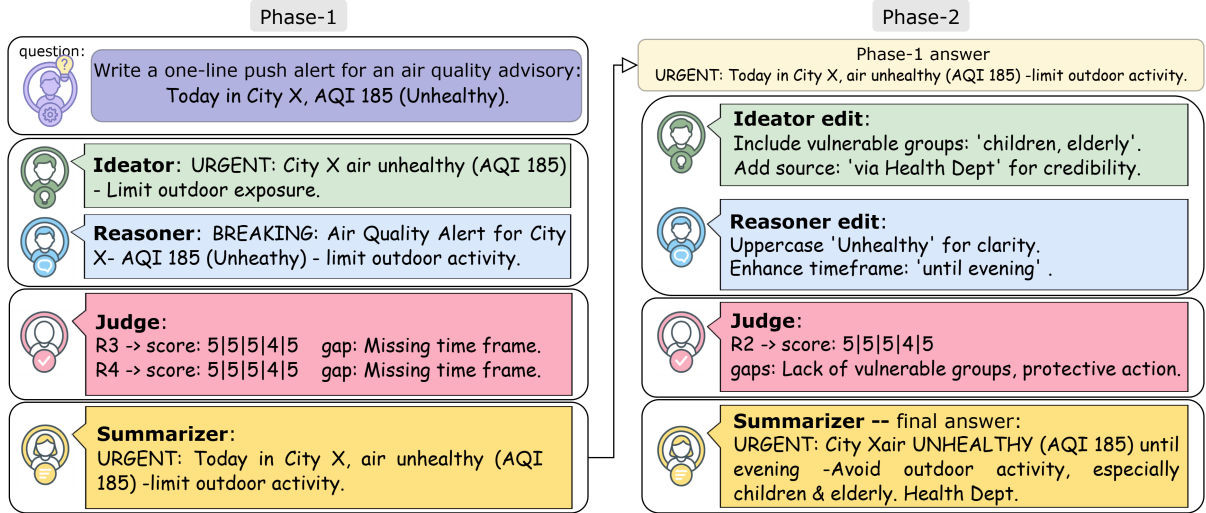


Figure 1: Case example of the two-phase RS-MA pipeline. In Phase 1, the Ideator and Reasoner draft candidates, the Judge identifies gaps, and the Summarizer produces an initial answer. In Phase 2, targeted edits and re-judgment guide the final synthesis.

edits, and final integration. Intermediate judgments and revisions are stored in JSON, which makes the process easier to inspect than free-form debate. Figure 1 illustrates this division of labor on a representative example about air quality.

RS-MA also includes an audition-based assignment procedure. Candidate models answer role-aligned prompts, rankings are aggregated with Borda scores, and a one-to-one Hungarian assignment provides an interpretable initial role map. We treat this map as a stable assignment prior rather than a claim of global optimality. We then evaluate alternative role maps under the same protocol, because the best deployment map can depend on the benchmark and on available model endpoints.

On AlpacaEval 2.0, the audition-selected RS-MA reaches 88.86% length-controlled win rate, while a validated role map reaches 90.72%. The full 805-example role-assignment ablation shows that single-model reuse and some random assignments obtain high raw win rates only with much longer outputs. This supports the central claim that role specialization should be judged under length-controlled evaluation, not raw preference alone. RS-MA also remains competitive on MT-Bench and FLASK, indicating that the framework is not limited to a single automatic judge.

Our contributions are:

- We propose RS-MA, a prompting-only framework that assigns heterogeneous LLMs to interpretable roles in a two-phase refinement process at inference time.

- We introduce audition-based role assignment and validate it with full assignment ablations, showing that role maps materially affect length-controlled instruction following.

- We provide calibrated evidence across AlpacaEval 2.0, MT-Bench, FLASK, and ArenaHard v2.0, including uncertainty-aware AlpacaEval results and controls for verbosity-driven gains in preference evaluation.

2 Related Work

Recently, research on multi-agent systems (MAS) has expanded rapidly, spanning both theoretical foundations and practical coordination mechanisms. Building on classic paradigms in cooperative, competitive, and decentralized learning, this line of work highlights how diversity can be harnessed through coordination and credit assignment (e.g., value decomposition and centralized-critic training) (Sunehag et al., 2017; Lowe et al., 2017; Rashid et al., 2020; Foerster et al., 2018; Guo et al., 2024). This perspective motivates role assignment as a coordination problem for LLM systems.

Within the LLM era, collaboration methods can be organized into several related families. A recent survey organizes LLM-agent systems around construction, collaboration, and evolution (Luo et al., 2025). Role-based dialogue frameworks such as CAMEL, AutoGen, AgentVerse, and OpenAgents use explicit profiles and conversation programs (Li et al., 2023; Wu et al., 2023; Chen et al., 2024b;

Xie et al., 2023). Workflow-based and search-based systems such as HALO, MASTER, MAGIS, AgileCoder, OpenHands, and SWE-Agent coordinate specialized agents through structured trajectories, planning, or execution environments (Hou et al., 2025; Gan et al., 2025; Tao et al., 2024; Nguyen et al., 2025; Wang et al., 2025c; Yang et al., 2024). Debate and consensus methods such as MAD and Reconcile elicit competing views before adjudication (Liang et al., 2024; Chen et al., 2024a). Aggregation methods such as LLM-Blender and Mixture-of-Agents rank, fuse, or layer candidate responses (Jiang et al., 2023; Wang et al., 2025b). Decentralized designs such as Symphony further study collective behavior under heterogeneous deployments (Wang et al., 2025a). HEAL applies agent collaboration to text-attributed hypergraph learning (Li et al., 2025b), beyond response generation.

Meanwhile, prompting and scaling strategies for a single LLM continue to improve inference. These advances include techniques that make intermediate reasoning explicit or broaden the search process, such as Chain-of-Thought, Self-Consistency, and Tree/Graph-of-Thoughts (Wei et al., 2022; Wang et al., 2022; Yao et al., 2023a; Besta et al., 2024), as well as methods that interleave reasoning with actions for controllable tool use, notably ReAct (Yao et al., 2023b). Separately, conditional computation with sparsely gated Mixture-of-Experts and automatic sharding (e.g., GShard) scales a single model by routing tokens to a small subset of internal experts. This increases model capacity without a proportional increase in per-token computation. Although related in spirit, such mechanisms are complementary to multi-agent orchestration and remain conceptually distinct (Shazeer et al., 2017; Lepikhin et al., 2020). We therefore use strong single-model systems as baselines rather than substitutes for cross-model role assignment.

Evaluation research has developed benchmarks that emphasize different aspects of model behavior. MT-Bench and Chatbot Arena assess multi-turn dialogue and pairwise preference while exposing position and length effects in model-based judging (Zheng et al., 2023). Length-Controlled AlpacaEval reduces length sensitivity in instruction-following comparisons (Dubois et al., 2024), while Arena-Hard extends automatic evaluation to 500 challenging open-ended prompts curated from large conversation datasets (Li et al., 2024). FLASK complements these preference benchmarks by decomposing overall performance

into twelve fine-grained skills (Ye et al., 2024). A-OKVQA adds a multimodal setting that requires commonsense and world knowledge beyond the paired image (Schwenk et al., 2022). Recent analyses further examine position bias and broader judge behavior (Shi et al., 2025; Li et al., 2025a). Together, these benchmarks provide complementary views of dialogue quality, prompt difficulty, skill composition, and knowledge-intensive visual reasoning across benchmark settings.

Taken together, prior work establishes strong foundations for leveraging heterogeneity and tooling in multi-agent orchestration. Bias-aware role design, principled role-model assignment, and length-robust aggregation that preserves attribution and verifiability remain underexplored. In particular, existing systems often rely on ad hoc role definitions or fixed model choices, and their aggregation can be sensitive to verbosity, making it harder to ensure fair comparison and faithful credit assignment across settings. Our framework directly targets these gaps by linking role selection, evaluation, and refinement in one protocol.

3 Methodology

We arrange heterogeneous LLMs into four roles that communicate via a structured text-only protocol: **Judge (R1)** for conservative, rule-sensitive evaluation and gap discovery, **Summarizer (R2)** for compression, re-organization, and coherence, **Ideator (R3)** for diverse and concrete proposal generation, and **Reasoner (R4)** for stepwise, constraint-aware reasoning. Given an input query x , the system executes a two-phase refinement process, with each phase producing explicit intermediate artifacts. Figure 2 gives an overview. All role-level system prompts are listed in Appendix A, matching the formal operators defined below.

3.1 Phase-1: Candidate Generation, Judging, and Structured Synthesis

Phase 1 aggregates complementary candidates for the same input x in complementary styles and emphases. The Ideator operator $\mathcal{I}$ is instantiated by a system prompt that requires covering core points while allowing novel angles and comparisons. It must include 1-3 short concrete examples, avoid buzzwords, and not fabricate data or sources. The Ideator output is defined as

$$y^{(3)} = \mathcal{I}(x). \quad (1)$$

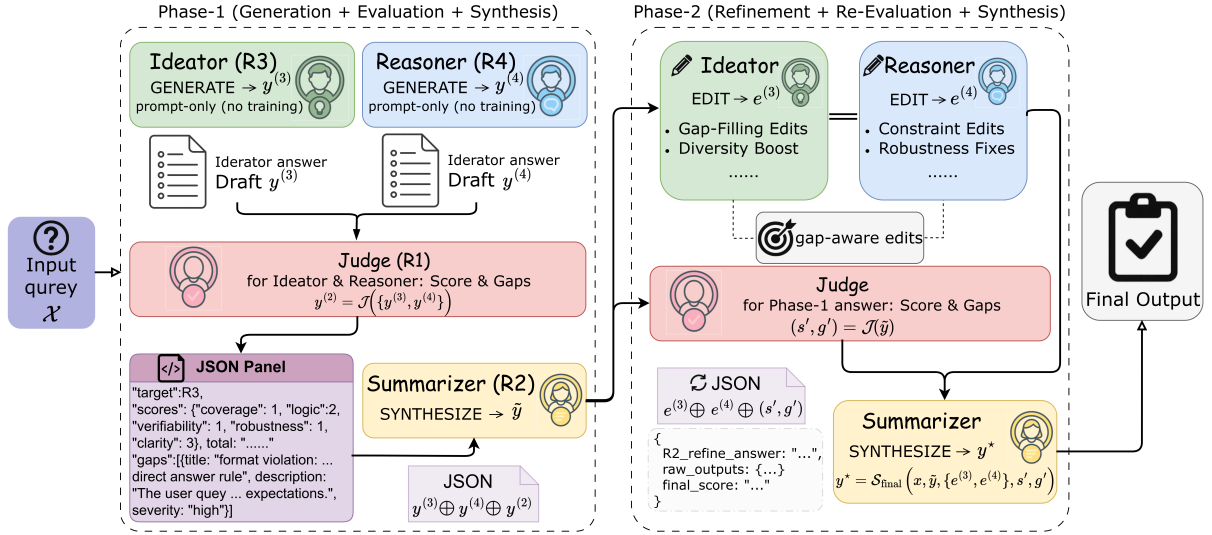


Figure 2: Overview of the RS-MA framework. In Phase 1, the Ideator and Reasoner draft answers, the Judge scores them and surfaces gaps, and the Summarizer consolidates them into a JSON-logged draft. In Phase 2, the Judge re-evaluates the Phase 1 answer, the Ideator and Reasoner provide targeted edits, and the Summarizer integrates the draft, edits, and feedback.

In practice, diversity is shaped at prompt level (explicitly soliciting alternatives, styles, and examples), which emulates a higher sampling temperature $\tau_{\mathcal{I}}$ and a beam budget $K_{\mathcal{I}}$ without changing decoding internals or model weights.

The Reasoner operator $\mathcal{R}$ is prompted to produce a rigorous, verifiable structure: it begins with step-by-step reasoning or key bullets, states assumptions and constraints explicitly, and attaches boundary conditions to conclusions. Unverifiable facts are disallowed. We write

$$y^{(4)} = \mathcal{R}(x). \quad (2)$$

Outputs use a visible step format (numbered steps and assertions), for inspection and downstream checking. Downstream modules consume only the visible text, which keeps the interface auditable.

Both candidates are evaluated by the Judge. The Judge is instantiated to grade only (never write the final answer), returning a single JSON object with schema $\{\text{target, scores, total, gaps, overall_comment}\}$. Here, the field *scores* contains five 0-5 aspect scores (coverage, logic, verifiability, robustness, clarity), *total* is a scalar computed from these five aspects using a fixed weighted rubric on a 0-100 scale, *gaps* is a list of title, description, severity where severity is *low/medium/high* (integers 1/2/3 also accepted), and *overall_comment* is a single summary evaluation paragraph.

Let $\mathbf{r}(x, y^{(3)}, y^{(4)}) \in [0, 5]^5$ denote the five aspect scores, and let $\mathbf{w} \in \mathbb{R}_{\geq 0}^5$ be the aspect weights

with $\sum_{k=1}^5 w_k = 1$. The Judge total s is defined as

$$s = 100 \sum_{k=1}^5 w_k \frac{r_k}{5} = \mathbf{w}^\top (20 \mathbf{r}), \quad (3)$$

which linearly maps each aspect from $[0, 5]$ to $[0, 100]$ and averages by $\mathbf{w}$. The Judge JSON's gaps (typed with severities) are carried forward to guide Phase-2 edits.

Given (s, g) and the two candidates, the Summarizer computes an intermediate synthesis $\tilde{y}$ under a set of prompt-level constraints: use only the information already provided (i.e., introduce no new facts), unify wording and tone across candidates, prioritize content according to the Judge scores, and systematically address each gap listed in g . In implementation, we operationalize these requirements as a checklist-style synthesis instruction to keep $\tilde{y}$ coherent and strictly grounded in the provided candidates. Concretely, the Summarizer iterates over the gaps in g , merges supporting evidence from the candidates, and performs a final consistency pass to ensure that every included claim is traceable to the inputs and that the overall response fits within the length budget.

$$\tilde{y} = \mathcal{S}(x, \{y^{(3)}, y^{(4)}\}, s, g). \quad (4)$$

We view synthesis as a constrained aggregation procedure conducted under a fixed length budget $\mathcal{Y}_L$, and we make these constraints explicit by

Algorithm 1 Two-Phase RS-MA: Generation, Evaluation, and Refinement

```

1: Input: Query  $x$ , role models  $\{R_1, R_2, R_3, R_4\}$ 
2: Output: Final answer  $y^*$ 
3:  $y^{(3)} \leftarrow R_3.\text{GENERATE}(x)$ 
4:  $y^{(4)} \leftarrow R_4.\text{GENERATE}(x)$ 
5:  $(s, g) \leftarrow R_1.\text{JUDGE}(\{y^{(3)}, y^{(4)}\})$ 
6:  $\tilde{y} \leftarrow R_2.\text{SYNTHESIZE}(x, \{y^{(3)}, y^{(4)}\}, s, g)$ 
7:  $e^{(3)} \leftarrow R_3.\text{EDIT}(x, \tilde{y})$ 
8:  $e^{(4)} \leftarrow R_4.\text{EDIT}(x, \tilde{y})$ 
9:  $(s', g') \leftarrow R_1.\text{REJUDGE}(\tilde{y})$ 
10:  $y^* \leftarrow R_2.\text{SYNTHESIZEFINAL}(x, \tilde{y}, \{e^{(3)}, e^{(4)}\}, s', g')$ 
11: return  $y^*$ 

```

incorporating gap-aware requirements directly into the synthesis specification:

$$\tilde{y} = \arg \max_{y \in \mathcal{Y}_L} \{U(y, x) - G(y, g) - N(y)\}, \quad (5)$$

where U promotes coherence, organization, and completeness, $G(y, g)$ measures unresolved gaps, and $N(y)$ penalizes unsupported additions. In practice, these terms are operationalized through checklist instructions and self-critique patterns, without introducing a learned ranker.

3.2 Phase-2: Targeted Edits, Re-judging, and Final Integration

Given the Phase-1 synthesis $\tilde{y}$, the Ideator (R3) proposes localized edits to diversify the presentation and increase concreteness, for example by inserting 1–3 concise illustrative examples or alternative framings, while strictly avoiding the introduction of new facts and keeping changes strictly localized. The Reasoner (R4) proposes edits that strengthen step-by-step logic, make assumptions and constraints explicit, add boundary checks, and remove unverifiable claims. Under the same input x , the edits are instantiated as

$$e^{(3)} = \mathcal{I}_{\text{edit}}(x, \tilde{y}), \quad e^{(4)} = \mathcal{R}_{\text{edit}}(x, \tilde{y}), \quad (6)$$

These edits must address a non-empty subset of the previously identified gaps g without unsupported claims or changes to the evidence available at this stage of processing.

The Judge re-evaluates $\tilde{y}$ using the same judging procedure as in Phase-1, so as to refresh the corresponding scores and the identified gaps under the unchanged Phase 1 rubric:

$$(s', g') = \mathcal{J}(\tilde{y}), \quad (7)$$

and the Summarizer performs a second-round revision under potentially conflicting role feedback, explicitly prioritizing safety and the Judge-identified

gaps whenever suggestions from different roles conflict with one another:

$$y^* = \mathcal{S}_{\text{final}}(x, \tilde{y}, \{e^{(3)}, e^{(4)}\}, s', g'). \quad (8)$$

If any role returns a filtered/safety response, the pipeline emits a safe fallback message and terminates early, persisting intermediate artifacts to disk. This conservative fail-closed design also preserves a trace for debugging and audit after failed or filtered runs for subsequent inspection.

3.3 Model Evaluation and Assignment

We first select a distinct model for each role from a candidate pool $\mathcal{M}$ by combining rank-aggregated evidence across role-aligned aspects and solving a global one-to-one assignment. This assignment step exploits heterogeneous strength profiles while reducing correlated errors and stylistic lock-in across roles. We use it as an interpretable initial assignment, not as a guarantee that the resulting map is globally optimal for every downstream benchmark. Overall, we aggregate rankings via Borda (Saari, 1985) and then perform a linear assignment solved by the Hungarian algorithm (Kuhn, 1955). This keeps the assignment step deterministic once audition scores are fixed.

Before running the collaboration loop, we first assess each candidate model on four role-aligned aspects, so that each role can be instantiated with a model whose strengths are consistently aligned with its intended function: creativity, logical rigor, summarization/abstraction, and cautious judgment. For every aspect we design five short audition prompts (see Appendix B). Each model answers all prompts under multiple random seeds, and a fixed proprietary judge model grades every answer with the audition rubric, producing per-answer criterion scores in $\{0, 1, 2\}$ and an overall score in $[0, 10]$. For a given run (seed), we average the overall scores across the five prompts. Across seeds we compute the cross-seed mean

$$M_{i,a} := \frac{1}{S} \sum_{s=1}^S \phi_{a,s}(i), \quad (9)$$

where $\phi_{a,s}(i)$ is the per-run mean overall (in $[0, 10]$) for model i on aspect a under seed s .

Borda Rank Aggregation Within each aspect a , we convert per-run rankings to Borda points using the $1, \dots, N$ convention with tie-average:

$$b_{i,a}^{(r)} = N - \text{rank}_{i,a}^{(r)} + 1, \quad (10)$$

where N is the number of candidate models. We aggregate the resulting Borda points across the R evaluation runs (i.e., judge seeds) to obtain a stable estimate that is less sensitive to incidental variation in any single assignment run.

$$B_{i,a} = \sum_{r=1}^R b_{i,a}^{(r)}. \quad (11)$$

The aspect-wise Borda scores $B_{i,a}$ form a utility matrix over models and roles, capturing complementary strengths across aspects and providing a common basis for comparing candidates under each role-aligned criterion.

We select distinct models for the four roles by maximizing the summed Borda under one-to-one constraints, which both prevents degenerate reuse of a single model across roles and aligns heterogeneous inductive biases with heterogeneous role demands in our setting, thereby encouraging complementary specialization rather than correlated failure modes across the overall collaborative pipeline. In practice, this formulation reduces the role-assignment problem to selecting exactly one model per role from $\mathcal{M}$ while ensuring each model is used at most once:

$$\begin{aligned} \max_{X \in \{0,1\}^{|\mathcal{M}| \times 4}} \quad & \sum_{i \in \mathcal{M}} \sum_{a \in \mathcal{A}} B_{i,a} X_{i,a} \\ \text{s.t.} \quad & \sum_{a \in \mathcal{A}} X_{i,a} \leq 1, \forall i \in \mathcal{M}, \\ & \sum_{i \in \mathcal{M}} X_{i,a} = 1, \forall a \in \mathcal{A}, \end{aligned} \quad (12)$$

where $X_{i,a} \in \{0,1\}$ indicates whether model i is assigned to role a , $\mathcal{M}$ is the candidate set, and $\mathcal{A} = \{1, 2, 3, 4\}$ (Judge, Summarizer, Ideator, Reasoner). Tied ranks are converted to Borda points via tie-average before the global assignment. The one-to-one optimization resolves competition across roles, reduces correlated failure modes, and leverages complementary peaks in $B_{i,a}$. We solve Eq. (12) using the Hungarian algorithm in our experiments, which makes the selected map reproducible from the audition table.

Validated role maps. The audition procedure gives a reproducible and interpretable assignment prior. For the final deployment analysis, we also evaluate a small set of role maps under the same benchmark and evaluator. This validation step is important because a model can be useful in more than one role, and endpoint changes can alter the practical trade-off between complementarity, length,

Table 1: AlpacaEval 2.0 results. LC win is the primary metric for model comparison.

Model	LC win	Raw win	Length
Qwen-3	83.58	83.77	2310
Kimi-K2	83.08	74.30	1867
DeepSeek-V3.1	76.31	88.28	3195
GPT-5	69.03	73.52	3052
GLM-4.5	67.78	57.72	1796
MoA (open-source)	65.10	59.80	1825
ERNIE-4.0-T	44.75	44.16	1910
LLaMA-3.3	34.40	22.20	1919
RS-MA, audition map	88.86	88.28	2259
RS-MA, validated map	90.72	90.69	2375

and raw preference. We therefore report both the audition-selected RS-MA and the best validated map in Sec. 4.3.

4 Experiments

4.1 Settings

Benchmarks. We evaluate RS-MA on AlpacaEval 2.0 (Dubois et al., 2024), MT-Bench (Zheng et al., 2023), and FLASK (Ye et al., 2024), as well as the 500-query hard-prompt split of Arena-Hard v2.0 (Li et al., 2024). AlpacaEval reports raw win rate and length-controlled (LC) win rate. We use LC win as the primary metric because it reduces the effect of verbosity in automatic preference judgments. MT-Bench contains 80 two-turn questions across eight categories and is scored on a 0 to 10 scale. FLASK decomposes instruction following into twelve skill dimensions. For Arena-Hard v2.0, we use the official GPT-4.1 judge in both answer orders and report the benchmark’s style-controlled score, which adjusts for response length and Markdown formatting during final scoring.

Baselines and variants. For AlpacaEval, we compare against strong single-model systems and an open multi-agent reference, including GPT-5, Qwen-3, Kimi-K2, DeepSeek-V3.1, GLM-4.5, ERNIE-4.0-T, LLaMA-3.3, and MoA. Single-model baselines test orchestration against one model, while MoA tests multi-model aggregation without role assignment or Phase 2. Role-map variants and Phase-1-only are within-method ablations. The audition-selected RS-MA uses Qwen-3 as Judge, Kimi-K2 as Summarizer, GLM-4.5 as Ideator, and DeepSeek-V3.1 as Reasoner. For role-assignment analysis, we additionally evaluate validated, random, and single-model maps on the full 805-example AlpacaEval set. All other results use the audition map by default.

Table 2: MT-Bench results. Scores are judged on a 0 to 10 scale for each turn.

Model	Avg.	Turn 1	Turn 2
MoA (open-source)	9.25	9.44	9.07
GPT-5	9.23	9.00	9.45
LLaMA-3.3	8.94	9.20	8.68
Qwen-3	9.31	9.19	9.44
DeepSeek-V3.1	9.20	9.29	9.11
GLM-4.5	7.31	7.01	7.60
ERNIE-4.0-T	8.84	8.76	8.93
Kimi-K2	9.17	8.98	9.36
RS-MA	9.48	9.51	9.45

4.2 Main Results

AlpacaEval 2.0. Table 1 reports AlpacaEval results for strong single-model baselines, an open multi-agent reference, and RS-MA. The audition-selected RS-MA reaches 88.86 LC win. A validated role map further improves LC win to 90.72, while keeping the average answer length far below the long single-model controls in Sec. 4.3. We therefore treat the validated map as the strongest AlpacaEval deployment, and the audition-selected map as the interpretable default. The LC and raw split shows that the gain is not a length effect alone, since long single-model answers can improve raw preference while losing after length control, a pattern examined directly in the role-assignment ablation. DeepSeek-V3.1 provides the clearest length contrast, with 88.28 raw win, 76.31 LC win, and an average length of 3195, while the validated map reaches 90.72 LC at an average length of 2375.

MT-Bench. RS-MA remains competitive on MT-Bench and attains the highest average score in our cached evaluation (Table 2). Compared with MoA, RS-MA raises the average from 9.25 to 9.48 and improves both turns. Against Qwen-3, the gain comes mainly from Turn 1, while Turn 2 remains nearly unchanged at 9.45 versus 9.44.

FLASK. Figure 3 reports the twelve-skill FLASK profile. The numeric aggregation gives RS-MA a mean score of 4.58, an average rank of 1.67, and the best score on 8 of 12 skills. The strongest gains appear in skills that benefit from explicit checking and synthesis, including factuality, robustness, and insightfulness. This profile is consistent with the role design, where checking, candidate generation, and synthesis are separated before the final response is produced. Appendix C reports A-OKVQA results with multimodal Ideator and Reasoner models.

Arena-Hard v2.0. On the 500-query hard-prompt split of Arena-Hard v2.0 (Li et al., 2024), full RS-

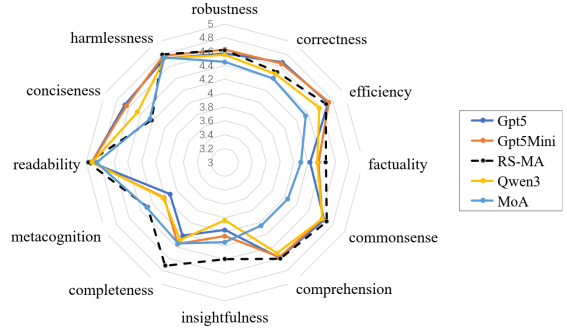


Figure 3: FLASK radar over twelve skills. RS-MA has a balanced profile and wins 8 of 12 skill dimensions in the numeric aggregation.

Table 3: Arena-Hard v2.0 hard-prompt results under the official GPT-4.1 style-controlled evaluation.

System	Style-controlled score	Bootstrap interval
GPT-4.1	58.3	-2.0/ + 2.3
Qwen-3	57.1	-2.0/ + 1.6
Kimi-K2	54.5	-1.9/ + 2.2
o3-mini	50.0	-0.0/ + 0.0
DeepSeek-V3.1	44.2	-2.0/ + 1.9
GLM-4.5	40.9	-1.6/ + 2.0
RS-MA	77.8	-1.6/ + 1.5

MA scores 77.8 with a 95% bootstrap interval from 76.2 to 79.3 under the protocol described above. The bundled o3-mini reference scores 50.0, while the strongest component, Qwen-3, scores 57.1.

The 20.7-point gain over Qwen-3, the strongest component, indicates that RS-MA benefits from the coordinated use of multiple component models through the two-phase workflow.

4.3 Role Model Selection and Assignment

We first summarize the audition outcomes used to obtain the interpretable default map. The candidate pool contains six production LLMs: Qwen-3, Kimi-K2, GLM-4.5, DeepSeek-V3.1, ERNIE-4.0-T, and LLaMA-3.3. Figure 4 shows the per-role Borda landscape. Different roles peak at different models, which supports heterogeneous assignment.

Across 10 audition seeds, the full four-role mapping is recovered in 8 runs. R3 and R4 are stable in all 10 runs. The remaining variation is limited to a near-tie between the top candidates for R1 and R2. Table 4 reports the selected model for each role.

Leave-One-Prompt-Out Analysis. We assess prompt sensitivity by removing each of the 20 audition prompts once. For each deletion, we keep

Table 4: Audition-selected role assignment. Mean is the cross-seed average overall score for the selected model.

Role	Selected model	Borda	Mean
R1 Judge	Qwen-3	20.5	8.675
R2 Summarizer	Kimi-K2	22.5	9.700
R3 Ideator	GLM-4.5	15.0	9.850
R4 Reasoner	DeepSeek-V3.1	19.5	9.000

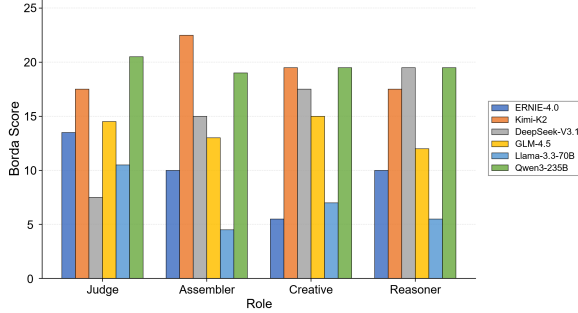


Figure 4: Borda scores across candidate models by role. Different peaks support heterogeneous assignment.

the six candidate models, the other 19 prompts, the recorded judgments, and all other settings fixed, then recompute score aggregation, Borda ranks with tied ranks averaged, and the optimal one-to-one assignment. The full prompt set reproduces the submitted assignment with a total Borda score of 77.5. The submitted map remains the unique optimum in 18 of 20 deletion conditions, or 90%, and an optimal assignment retains the same four-model set in 19 conditions, or 95%. The exact assignment changes after removing one Judge prompt or one Reasoner prompt, while only the Judge case changes the selected model set. The Summarizer and Ideator remain unchanged across all deletions. Across the four roles, affected-role mean Kendall τ_b over the five corresponding deletions ranges from 0.762 to 0.936. Together, the stable model set and role rankings indicate that the assignment is supported by evidence across the prompt set rather than by one decisive prompt.

Figure 4 shows how one-to-one assignment changes the role-wise choice. Qwen-3 leads the Judge scores and ties for the Ideator and Reasoner roles, while Kimi-K2 leads the Summarizer scores and ties for the Ideator role. Joint optimization under the one-model-per-role constraint yields the map reported in Table 4.

Table 5 lists the role maps used in the full AlpacaEval assignment ablation. The variants separate three explanations: role-specialized selection, arbitrary heterogeneous assignment, and reuse of

Table 5: Role maps in the full assignment ablation.

Variant	R1	R2	R3	R4
Audition map	Qwen	Kimi	GLM	DeepSeek
Random-s3	GLM	Kimi	Qwen	DeepSeek
Validated map	Qwen	Kimi	Kimi	DeepSeek
Random-s6	DeepSeek	GLM	Kimi	Qwen
Single Qwen	Qwen	Qwen	Qwen	Qwen

Table 6: Full AlpacaEval role-assignment ablation. LC SE denotes official length-controlled standard error.

Variant	LC $\pm$ SE	Raw	Length
Audition map	88.86 $\pm$ 0.12	88.28	2259
Random-s3 map	89.59 $\pm$ 0.14	90.34	2415
Validated map	90.72 $\pm$ 0.08	90.69	2375
Random-s6 map	85.53 $\pm$ 0.16	94.82	4168
Single Qwen map	85.79 $\pm$ 0.15	95.36	4042

one strong backbone across all roles.

The full assignment ablation in Table 6 tests whether this assignment step matters. The validated map, which uses Kimi-K2 for both R2 and R3, obtains the best LC win at 90.72. This result changes the interpretation of the assignment module. The one-to-one audition map is a stable and interpretable default, while benchmark-side role-map validation can identify a stronger deployment map when reuse of a model across roles is allowed.

The random controls show that assignment quality is not reducible to model strength alone. Random-s3 is competitive, which suggests that the Summarizer and Reasoner choices carry much of the benefit in this comparison. Random-s6 gives the clearest counterexample to a simple heterogeneity story. It uses four different models, yet it loses more than five LC points relative to the validated map. The long-output controls are also informative. Random-s6 and single Qwen obtain raw win rates above 94, but their LC wins fall below 86 because their average lengths exceed 4000 tokens. The validated map improves LC without this length inflation, which is the central evidence for the validated deployment map on this benchmark.

For transparency, random-s3 and random-s6 include 13 and 1 conservative fallback rows after endpoint failures, so those two results should be read conservatively. The validated map and the single-Qwen map do not use fallback rows, so the main comparisons are not affected.

Table 7: Phase-2 trade-off on full AlpacaEval 2.0.

Variant	LC win	Raw win	Length saving
Phase-1-only	86.79	77.61	NA
Selective Phase 2	88.77	83.83	13.96%
Full RS-MA	88.86	88.28	0.00%

Table 8: Full AlpacaEval 2.0 deployment trade-off. Calls are average model calls per query.

Method	LC win	Calls	Avg. length	Phase-2 rows
Phase-1-only	86.79	5.00	1655	0/805
Selective Phase 2	88.77	7.36	1944	474/805
Full RS-MA	88.86	9.00	2259	805/805

4.4 Ablation Study

We next separate role assignment from the two-phase refinement loop. On the full matched AlpacaEval evaluation, Phase-1-only reaches 86.79 LC and 77.61 raw win, while full RS-MA reaches 88.86 LC and 88.28 raw win. This supports a narrower claim: Phase 2 is not merely an LC optimizer. It improves completeness and raw preference through re-judgment and targeted edits, while sometimes increasing length. Appendix D shows the added boundary conditions and mechanisms.

We also evaluate Selective Phase 2, which uses task type and instruction length to decide whether each query should enter the refinement stage. The policy runs Phase 1 for every query and applies Phase 2 only to the selected subset. Selective Phase 2 reaches 88.77 LC and 83.83 raw win while reducing output length by 13.96% relative to full RS-MA (Table 7). The small LC difference indicates that many second-pass edits improve completeness more than length-normalized preference, while the larger raw-win gap shows why the second pass remains useful when users value complete answers. The 10.67-point raw gain from Phase-1-only to full RS-MA further shows that the second pass is most useful when completeness is central.

4.5 Cost and Deployment Trade-Off

The full 805-example results also show the practical cost-quality trade-off. RS-MA is most useful when answer quality and traceable intermediate feedback justify higher inference cost. For cost-sensitive settings, the selective policy runs the second phase only on examples selected by the gate.

Tables 7 and 8 show this trade-off. Full RS-MA gives the strongest raw preference and runs Phase 2 for every query. The selective Phase 2

policy keeps a similar LC win rate while reducing second-phase executions to 474 of 805 examples. It better fits offline or audit-sensitive settings than low-latency chat deployment. For high-throughput serving, the evidence favors the selective Phase 2 policy or a single strong model unless the quality margin justifies latency and cost.

5 Conclusion

We presented RS-MA, a prompting-only framework for heterogeneous LLM orchestration with interpretable roles. Full assignment ablations show that role maps affect length-controlled instruction following, and the validated map improves AlpacaEval LC without the length inflation of single-model or random controls. Phase 2 improves completeness and raw preference, while its selective variant reduces model calls and output length. MT-Bench and FLASK show strong dialogue and skill profiles, while Arena-Hard shows that the full pipeline outperforms its components. Overall, the results connect role assignment and refinement with deployment cost and inspectable evidence.

6 Limitations

RS-MA improves instruction following through inference-time orchestration, so it requires more model calls and API cost than a single model call. The validated role map is tied to the evaluated model pool and should be revalidated when available models, prices, or API behavior change. Our evidence comes from AlpacaEval 2.0, MT-Bench, FLASK, and Arena-Hard v2.0, which cover instruction following, dialogue, and fine-grained skills but not long-horizon tool use, multilingual robustness, or safety-critical deployment. Finally, our ablations evaluate role assignment at the system level and do not fully isolate interactions among model identity, prompt wording, and judge preference.

Acknowledgments

This work is supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2026ZD0125403, the National Major Scientific Instruments and Equipment Development Project of the National Natural Science Foundation of China under Grant 62427820, and the Ministry of Education Engineering Research Center Guiding Project for Machine Learning and Industrial Intelligence Applications under Grant SCU2024D013.

References

- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.
- Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. [ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024b. [Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors](#). In *International Conference on Learning Representations*, volume 2024, pages 20094–20136.
- Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2024. [Self-collaboration code generation via chatgpt](#). *ACM Trans. Softw. Eng. Methodol.*, 33(7).
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpaca-eval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Wenlong Fang, Qiaofeng Wu, Jing Chen, and Yun Xue. 2025. [Notes-guided mllm reasoning: Enhancing mllm with knowledge and visual notes for visual question answering](#). In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19597–19607.
- Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Bingzheng Gan, Yufan Zhao, Tianyi Zhang, Jing Huang, Li Yusu, Shu Xian Teo, Changwang Zhang, and Wei Shi. 2025. [MASTER: A multi-agent system with LLM specialized MCTS](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9409–9426, Albuquerque, New Mexico. Association for Computational Linguistics.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. [Large language model based multi-agents: A survey of progress and challenges](#). *Preprint*, arXiv:2402.01680.
- Zhipeng Hou, Junyi Tang, and Yipeng Wang. 2025. [Halo: Hierarchical autonomous logic-oriented orchestration for multi-agent llm systems](#). *Preprint*, arXiv:2505.13516.
- Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. 2023. [Promptcap: Prompt-guided image captioning for vqa with gpt-3](#). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2951–2963.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. [LLM-blender: Ensembling large language models with pairwise ranking and generative fusion](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.
- Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. [Pythia v0.1: the winning entry to the vqa challenge 2018](#). *Preprint*, arXiv:1807.09956.
- Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. 2022. Webly supervised concept expansion for general purpose vision models. In *Computer Vision – ECCV 2022*, pages 662–681, Cham. Springer Nature Switzerland.
- H. W. Kuhn. 1955. [The hungarian method for the assignment problem](#). *Naval Research Logistics Quarterly*, 2(1-2):83–97.
- Dmitry Lepikhin, HyounJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhat-tacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025a. [From generation to judgment: Opportunities and challenges of LLM-as-a-judge](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. [Camel: Communicative agents for "mind" exploration of large language model society](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 51991–52008. Curran Associates, Inc.
- Ruochang Li, Xiao Luo, Zhiping Xiao, Wei Ju, and Ming Zhang. 2025b. [Heal: Hybrid enhancement with llm-based agents for text-attributed hypergraph self-supervised representation learning](#). In *EMNLP (Findings)*, pages 6815–6829.

- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. [From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline](#). *Preprint*, arXiv:2406.11939.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.
- Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. [Multi-agent actor-critic for mixed cooperative-competitive environments](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, and 7 others. 2025. [Large language model agent: A survey on methodology, applications and challenges](#). *Preprint*, arXiv:2503.21460.
- Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. 2021. [Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa](#). In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14106–14116.
- Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. [Clipcap: Clip prefix for image captioning](#). *Preprint*, arXiv:2111.09734.
- Minh Huynh Nguyen, Thang Phan Chau, Phong X. Nguyen, and Nghi D. Q. Bui. 2025. [Agilecoder: Dynamic collaborative agents for software development based on agile methodology](#). In *2025 IEEE/ACM Second International Conference on AI Foundation Models and Software Engineering (Forge)*, pages 156–167.
- Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. [Monotonic value function factorisation for deep multi-agent reinforcement learning](#). *Journal of Machine Learning Research*, 21(178):1–51.
- Donald G Saari. 1985. The optimal ranking method is the borda count. Technical report, Discussion paper.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. [A-okvqa: A benchmark for visual question answering using world knowledge](#). In *Computer Vision – ECCV 2022*, pages 146–162, Cham. Springer Nature Switzerland.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). *arXiv preprint arXiv:1701.06538*.
- Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. [Judging the judges: A systematic study of position bias in LLM-as-a-judge](#). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2017. [Value-decomposition networks for cooperative multi-agent learning](#). *Preprint*, arXiv:1706.05296.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Wei Tao, Yucheng Zhou, Yanlin Wang, Wenqiang Zhang, Hongyu Zhang, and Yu Cheng. 2024. [Magis: Llm-based multi-agent framework for github issue resolution](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 51963–51993. Curran Associates, Inc.
- Ji Wang, Kashing Chen, Xinyuan Song, Ke Zhang, Lynn Ai, Eric Yang, and Bill Shi. 2025a. [Symphony: A decentralized multi-agent framework for scalable collective intelligence](#). *Preprint*, arXiv:2508.20019.
- Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. 2025b. [Mixture-of-agents enhances large language model capabilities](#). In *International Conference on Learning Representations*, volume 2025, pages 33944–33963.
- Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 5 others. 2025c. [Openhands: An open platform for](#)

- ai software developers as generalist agents. In *International Conference on Learning Representations*, volume 2025, pages 65882–65919.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Zhao Wang, Sota Moriyama, Wei-Yao Wang, Briti Gangopadhyay, and Shingo Takamatsu. 2025d. [Talk structurally, act hierarchically: A collaborative framework for llm multi-agent systems](#). *Preprint*, arXiv:2502.11098.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. [Autogen: Enabling next-gen llm applications via multi-agent conversation](#). *Preprint*, arXiv:2308.08155.
- Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, and Tao Yu. 2023. [Openagents: An open platform for language agents in the wild](#). *Preprint*, arXiv:2310.10634.
- John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. [Swe-agent: Agent-computer interfaces enable automated software engineering](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 50528–50652. Curran Associates, Inc.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). *Preprint*, arXiv:2210.03629.
- Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. [Flask: Fine-grained language model evaluation based on alignment skill sets](#). In *International Conference on Learning Representations*, volume 2024, pages 55361–55414.
- Zhou Yu, Xuecheng Ouyang, Zhenwei Shao, Meng Wang, and Jun Yu. 2025. [Prophet: Prompting large language models with complementary answer heuristics for knowledge-based visual question answering](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6797–6808.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A System Prompts for RS-MA Roles

This appendix reports the role prompts used by RS-MA. Prompts are shown in compact form for readability. The implementation uses the same role instructions and stores intermediate judgments and edits as JSON records.

R3 Ideator, generation. The Ideator is instructed to produce a diverse answer with concrete examples. It should cover the core points, add useful angles or comparisons, provide one to three short examples, avoid slogans, and avoid fabricated data or sources.

R4 Reasoner, generation. The Reasoner is instructed to produce a rigorous and verifiable answer. It should use stepwise reasoning or structured bullets, state assumptions and constraints, include boundary conditions for conclusions, and avoid unverifiable claims.

R1 Judge, grading. The Judge grades answers only and does not write the final response. It evaluates coverage, logic, verifiability, robustness, and clarity on a 0 to 5 scale. It returns a JSON object with fields for the target answer, scores, total score, gaps, and an overall comment. Each gap contains a title, description, and severity.

R2 Summarizer, Phase 1 synthesis. The Summarizer assembles a stronger answer from the R3 and R4 candidates plus the Judge scores and gaps. It is instructed to use only the given information, unify tone and wording, address Judge-identified gaps when possible, and avoid boilerplate.

R3 and R4, Phase 2 edit suggestions. During Phase 2, R3 suggests localized improvements that increase concreteness or presentation diversity, while R4 suggests edits that strengthen logic and boundary handling. Both roles return concise JSON suggestions rather than full rewrites.

R2 Summarizer, final synthesis. The final Summarizer integrates the Phase 1 draft, R3/R4 suggestions, and the second Judge review. When suggestions conflict, it prioritizes Judge-identified gaps, safety, and task constraints.

B Audition Prompt Set

The audition uses 20 short prompts, five for each role-aligned aspect. Each candidate model answers all prompts under multiple seeds. A fixed proprietary judge model scores each answer with the audition rubric described in Sec. 3.3.

R1 Judge Prompts

1. Identify the flaw in “All swans are white because the ones I have seen are white” and give one verification step.
2. A report claims that retention significantly improved from an A/B test with $n = 28$. State the main risk and one validation method.
3. A news article cites only one anonymous source. Explain the credibility issue and give one cross-check.
4. A model reports 100% test accuracy. Explain why this is insufficient and give one robustness check.
5. A voluntary survey reports 92% satisfaction. Name the sampling issue and one remedy.

R2 Summarizer Prompts

1. Turn notes about an unclear goal, many options, a tight timeline, incomplete data, and review needs into three to five actionable bullets.
2. Merge two short project plans into one unified statement without adding facts.
3. Compress a process into four steps: Prepare, Execute, Check, and Deliver.
4. Summarize a short source about data gaps, a quick pilot, a two-week prototype, and no extra headcount in 50 to 60 words.
5. Normalize inconsistent terms and produce a checklist with at most five bullets.

R3 Ideator Prompts

1. List five ways to improve remote meeting efficiency, with one use case per line.
2. Give five diagnostic directions for high new-user churn, each with one validation idea.
3. Propose five ideas to increase newsletter open rate, each with a testable metric.
4. Provide four onboarding experiments, each with a success criterion.
5. Give five levers to improve freemium-to-paid conversion, each with a risk note.

Table 9: A-OKVQA validation results. MC is multiple-choice accuracy and DA is direct-answer score.

Method	MC	DA
ClipCap (Mokady et al., 2021)	44.0	18.1
Pythia (Jiang et al., 2018)	49.0	25.2
ViLBERT (Lu et al., 2019)	49.1	30.6
LXMERT (Tan and Bansal, 2019)	51.4	30.7
KRISP (Marino et al., 2021)	51.9	33.7
GPV-2 (Kamath et al., 2022)	60.3	48.6
PromptCap (Hu et al., 2023)	73.2	56.3
Prophet (Yu et al., 2025)	76.4	58.2
NoteMR (Fang et al., 2025)	88.1	68.7
RS-MA	87.6	70.0

R4 Reasoner Prompts

1. Decide whether a string is a palindrome after normalizing case and non-alphanumeric characters. Provide steps and two assertions.
2. Explain why A/B tests can yield false positives and give one preventive check.
3. Draft a minimal rollback plan for a production change, including one rollback trigger.
4. Implement a token-bucket rate limiter and state one invariant.
5. Decide whether a directed graph has a cycle using DFS or Kahn’s algorithm, with two assertions.

C Additional Vision-Language Validation

Table 9 reports an additional vision-language validation on the A-OKVQA validation split (Schwenk et al., 2022). The goal is to test whether the same division of labor transfers beyond text-only instruction following. We instantiate the Ideator and Reasoner with GLM-4.5V and Qwen3-VL-235B-A22B-Instruct, while keeping the Judge and Summarizer as textual controllers. Following the official A-OKVQA protocol, MC is measured by multiple-choice accuracy, and DA uses the standard VQA direct-answer scoring.

D Qualitative Case Analysis

Figure 5 examines a climate-zones query that asks why tropical, temperate, and polar zones exist, what the limitations of this classification are, and how the zones may change under global warming. The query is useful for qualitative analysis because a strong answer must combine physical mechanisms, classification caveats, and future impacts. The comparison therefore probes whether RS-MA improves

only surface fluency or whether its role structure changes the kinds of information preserved in the final answer.

Boundary and exceptions. The full RS-MA answer explicitly frames climate zones as abstractions rather than physical boundaries and then names mixed cases such as mountainous regions and coastal polar-temperate blends. This distinction matters because the question asks not only for the origin of climate zones but also for the limitations of the classification. The variant without R1 instead presents the zones as a clearer latitude-based partition, while the Phase-1-only answer mentions mountains and oceans without making fuzzy edges a central limitation. This contrast illustrates the function of the Judge. By surfacing missing boundary conditions, R1 encourages the final synthesis to include caveats that are easy to lose in a concise answer.

Verifiability and specifics. The full RS-MA answer also adds more checkable content. It refers to concrete urban heat island behavior, Hadley-cell movement, and the possible role of AMOC. These details make the answer more specific than a general statement that warming shifts precipitation belts or causes regional change. The ablated answers remain broadly correct, but they provide fewer named mechanisms and fewer quantities that a reader could inspect or verify. This pattern is consistent with the intended role of Phase 2. The second pass does not simply lengthen the answer. It gives the system another opportunity to fill gaps identified by the Judge and to convert broad claims into more accountable statements.

Consistency and coverage. The lower panel shows a coverage difference across mechanisms, limitations, and impacts. Full RS-MA connects solar geometry, atmospheric circulation, land-ocean contrast, model limitations, microclimates, novel climate types, biodiversity, and human systems within a single response. The Phase-1-only answer covers several mechanisms but has looser transitions and drops some downstream impacts. The variant without R1 gives a reasonable mechanism-level answer, yet it underdevelops human and ecological consequences. These omissions help explain why a shorter answer can perform well under length-controlled evaluation while receiving weaker raw preference when completeness is rewarded.

Relation to the ablation results. The case should not be read as an additional benchmark. It is a repre-

Boundary & Exceptions	Verifiability & Specifics
RS-MA: Classification is a model, not a physical boundary; mountainous regions and coastal polar-temperate blends often defy crisp zoning. w/o R1: Climate zones can be divided by latitude into tropical, temperate, and polar . . . boundaries are generally clear. w/o Phase-2: Mountains and oceans influence climate . . . no explicit mention of fuzzy edges or mixed zones.	RS-MA: Urban heat islands (e.g., downtown Tokyo can be 3-5 °C warmer); warming may shift Hadley cells by 0.5-1° per decade and affect AMOC. w/o R1: Warming expands the tropics and shifts precipitation belts . . . no quantified examples or AMOC mention. w/o phase-2: Examples include regional changes . . . lacks concrete numbers or named mechanisms.
Consistency & Coverage	
RS-MA: Mechanisms (insolation, circulation, land-ocean contrast), limits (zones are models; microclimates; new hybrids), impacts (biodiversity & human systems). w/o Phase-2: Lists mechanisms and effects, but transitions are loose and some subtopics are dropped. w/o R1: Mechanisms and warming effects . . . limited discussion on human/ecological impacts.	

Figure 5: Qualitative comparison on a climate-zones query. Blue text marks stronger content in RS-MA, and red text marks missing or weaker content in the ablated variants.

sentative example that clarifies the behavior behind Table 7. Phase-1-only is concise and can therefore remain competitive under length-controlled scoring. Full RS-MA adds a targeted refinement pass, which increases coverage and specificity and is reflected in the larger raw-win gain. Selective Phase 2 follows the same interpretation by applying the refinement pass only to examples selected by the task-type and instruction-length gate. The qualitative pattern supports the narrower claim in Sec. 4.4: Phase 2 is most useful when users value completeness, traceable reasoning, and explicit treatment of exceptions.

E Use of AI Assistants

AI assistants were used to help improve the clarity, grammar, and formatting consistency of the manuscript. All scientific claims, experimental design choices, reported results, analyses, and conclusions were determined and verified by the authors.