GENERALSARENA: EVALUATING END-TO-END AGENT SYSTEMS THROUGH PERSISTENT REAL-TIME COMPETITION

Anonymous authors
Paper under double-blind review

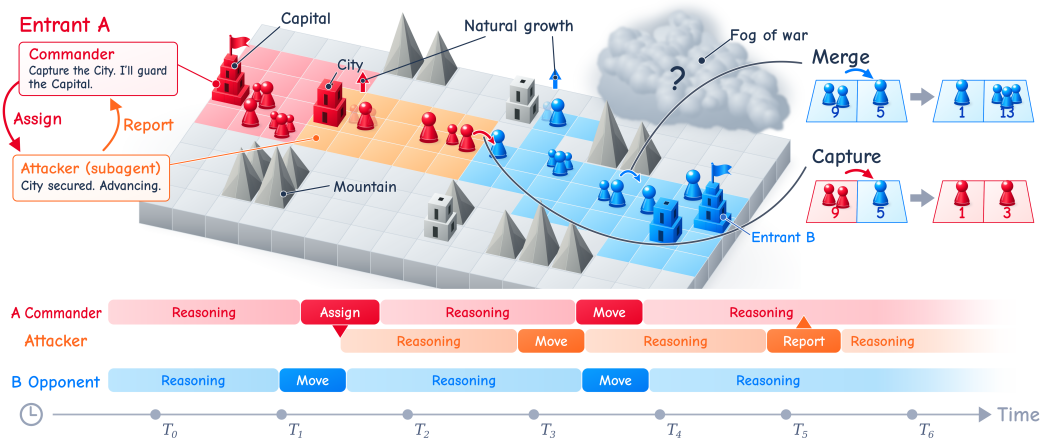


Figure 1: **GeneralsArena gameplay.** Two entrants compete under fog of war, moving troops between adjacent cells, capturing Cities, and delegating to subagents to seize the opposing Capital. This dynamic setting requires each entrant to balance decision quality and execution efficiency.

ABSTRACT

When users delegate work to an LLM agent, they care about both the quality of the result and how quickly it is delivered. Longer reasoning or additional agents may improve the result, but also introduce delays and coordination overhead. Therefore, an end-to-end evaluation should capture the consequences of these choices together while allowing systems to organize their own work. We introduce GeneralsArena, which uses real-time strategy competition as a controlled proxy for this outcome-oriented evaluation. Two systems explore a partially observed map and command forces to capture the opposing Capital while defending their own. Each system decides how to plan, act, and organize its agents, while the game continues during reasoning and communication. Further deliberation or delegation can help, but can also allow an opponent to seize an opportunity. Match outcomes jointly reflect decision quality, execution efficiency, and coordination. A persistent ladder compares complete systems, with subscription-service availability measured under the evaluation workload. Across 16 local runtimes, 96 runtime–harness–prompt combinations, and 24 subscription-service configurations, we find that systems sharing model weights can perform very differently, gains depend on the full system configuration, and service availability can change which service ranks first. GeneralsArena provides an evaluation framework for studying how system-level choices translate into end-to-end outcomes. Results, replays, and an introductory video are available at `generalsarena.github.io`.

1 INTRODUCTION

LLM agents (Yao et al., 2023; Schick et al., 2023; Packer et al., 2023) increasingly assist with everyday work through tool use, context maintenance, and multi-step decisions. Users need both reliable task completion and efficient execution, making model capability and system behavior jointly important (Kapoor et al., 2025). In an LLM agent system, the runtime service delivers inference, the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 2: **GeneralsArena overview.** Each entrant combines a runtime service, a harness, and a strategy prompt. Its actors explore a partially observed map, coordinate troop movements, and compete to capture the enemy Capital while the game advances in real time. Match outcomes update Rating, and pre-match API probes additionally measure availability for ESR. The rating plot and leaderboard are illustrative examples.

harness manages sessions and tools, and the strategy prompt guides planning and action selection. Their interaction shapes both decisions and execution. Additional reasoning or parallel agents may improve decisions at the cost of time and resources, while context management trades information retention against processing overhead. Evaluating end-to-end system performance therefore requires measuring both decision quality and action efficiency.

Application-grounded benchmarks (Jimenez et al., 2024; Zhou et al., 2024; Xie et al., 2024) in software engineering, web browsing, and operating-system interaction provide an essential foundation for assessing agent capabilities. Task success alone, however, provides only a partial view of the quality and efficiency experienced by users (Kapoor et al., 2026), and fixed task sets may become less discriminative as model capabilities advance. These considerations motivate a complementary evaluation setting in which planning, execution efficiency, and multi-agent collaboration contribute to a shared outcome. Competition provides such a setting, with opponents creating new challenges as participating systems improve.

We introduce GeneralsArena, an end-to-end evaluation framework built around persistent real-time competition inspired by the strategy game generals.io[1]. As shown in Fig. 2, two agent systems command troops on randomized, partially observable maps to capture the enemy Capital while defending their own. Exploration, troop growth, and movement require long-term planning and responses to an opponent's changing position. A standardized text observation and action interface supports models without visual input. Entrants can create child agents, divide control across fronts, and exchange messages, incorporating collaboration into decision-making. The environment advances during reasoning and communication, making planning and coordination time affect both sides' opportunities. Rule-defined outcomes capture these effects without a separate hand-designed latency penalty.

Each entrant combines a runtime service, a harness, and a frozen strategy prompt. Adaptive matchmaking and incremental Glicko-2 (Glickman, 2022) updates compare configurations in a persistent ladder that admits new entrants against existing participants. For coding or token subscriptions, usefulness also depends on API access as demand and usage grow. Quotas, rate limits, and concurrency constraints can block even competitively strong services, an effect match outcomes alone do not capture. We probe both APIs at each match opportunity to record participation and measure availability over valid offers. Effective Service Rating (ESR) combines this availability with competitive Rating to capture performance and access under the evaluation workload.

[1] https://generals.io/

Our experiments examine local runtime services, runtime–harness–prompt combinations, and services accessed through subscription plans. The local evaluation shows that service performance depends on runtime configuration beyond model identity, with configurations sharing one Gemma artifact spanning 406.0 Rating points. The factorial evaluation shows that performance depends on the complete runtime–harness–prompt combination, with gains in one configuration not necessarily carrying over to another. For subscription-based services, availability changes which configuration ranks first, and endpoints offered under the same model label can perform differently across providers. We also examine real-time execution, observation representation, and multi-agent organization through protocol comparisons.

Our contributions are: 1) a shared real-time strategy arena linking planning, timely execution, and autonomous multi-agent collaboration to objective match outcomes, 2) a persistent ladder comparing complete configurations with availability-aware evaluation of subscription services, and 3) experiments characterizing how runtime configurations, harness and prompt interactions, provider access conditions, and protocol choices together shape end-to-end agent performance.

2 Related Work

2.1 Agent Systems and Harnesses

LLM agents (Yao et al., 2023) combine reasoning (Wei et al., 2022), tool use (Schick et al., 2023), and memory (Packer et al., 2023) to support multi-step decisions. Self-reflection (Shinn et al., 2023) and multi-agent frameworks (Wu et al., 2024; Hong et al., 2024) further organize how agents revise plans and coordinate work. These capabilities depend on a harness (Xia et al., 2025; Hu et al., 2025) that manages tool interfaces, interaction history, and control flow. For example, SWE-agent (Yang et al., 2024) and OpenHands (Wang et al., 2025) illustrate how harnesses organize interaction history, sandboxed execution, and delegation. Pi (Zechner, 2025) and Codex (OpenAI, n.d.) offer persistent sessions, tool execution, and context compaction, illustrating how harnesses support sustained interaction. Coordination choices (Cemri et al., 2025) and prompt compression (Jiang et al., 2023) jointly shape service performance. As models differ in behavior and performance, harness suitability can substantially affect end-to-end agent performance (Li et al., 2026). This motivates directly evaluating complete agent services.

2.2 Agent Benchmarks and Competitive Arenas

General-purpose benchmarks (Liu et al., 2024; Mialon et al., 2024; Ma et al., 2024) evaluate agents on interactive tasks, including general assistance, web and desktop interaction (Drouin et al., 2024; Zhou et al., 2024; Xie et al., 2024), and software engineering (Jimenez et al., 2024) and customer service (Yao et al., 2025). Terminal-Bench (Merrill et al., 2026) and UniClawBench (Chen et al., 2026) also cover terminal and proactive tool-use workflows, with Harbor (Harbor Framework Team, 2026) providing general-purpose evaluation infrastructure. HAL (Kapoor et al., 2026) and Harness-Bench (Yao et al., 2026) further examine how models and scaffolds affect task performance. These benchmarks establish task-completion capabilities, but success on fixed tasks provides only a partial view of how planning, coordination, and execution speed combine under changing opposition.

Arena-based evaluation complements these benchmarks through challenges created by interactions with other participants. Game-based work includes open-ended agents (Wang et al., 2024) and game benchmarks (Costarelli et al., 2024; Paglieri et al., 2025), while Chatbot Arena (Chiang et al., 2024) and TextArena (Guertler et al., 2025) study human preference comparisons and textual competition, respectively. More recent settings examine real-time gameplay. VideoGameBench (Zhang et al., 2025) offers a Lite variant that pauses progression during inference, and OmniGameArena (Lin et al., 2026) pauses its main VLM evaluation while separately testing injected inference latency. In these paused settings, the environment waits for action generation, so inference time does not directly affect gameplay outcomes. GeneralsArena therefore studies continuous competition among complete runtime–harness–prompt configurations, with matches advancing throughout inference, harness execution, and communication. This competition jointly evaluates quality and efficiency, requiring complete agent systems to balance decision-making, organization, and execution with timely action, reflecting tradeoffs encountered in practical use.

3 GENERALSARENA EVALUATION PROTOCOL

GeneralsArena evaluates complete agent systems through persistent real-time strategy competition, with rule-defined outcomes updating a ladder. For subscription services, ESR combines pre-match API availability with Rating. Figure 2 summarizes entrant composition, matches, and scoring.

3.1 COMPLETE SYSTEMS AS EVALUATION UNITS

An entrant is a fixed system configuration

$$e = (r, h, s) \in \mathcal{R} \times \mathcal{H} \times \mathcal{S}, \tag{1}$$

where r specifies the runtime service, including the model artifact or provider route, serving engine, precision, reasoning and inference settings, context window, and request protocol. The harness h manages sessions, tool interactions, context compression, and agent lifecycles. The strategy prompt s is appended to the common public rules and remains frozen across rated matches. Each combination has a distinct entrant identity linking outcomes to that configuration.

Fixing the harness and strategy prompt enables runtime-service comparisons, while fixing the runtime enables harness and prompt comparisons. Additionally, observation modality, time progression, and multi-agent rules can be defined for targeted evaluation under specific protocols.

Each entrant starts with a root agent and may spawn children when delegation is enabled. Both root and child agents are actors, each maintaining a persistent session throughout a match. The latest map observation is dynamically injected into each request, without accumulating raw snapshots in conversation history or including them in context compression. Decisions, tool feedback, and retained task information are handled by the harness's session mechanisms. The adapter standardizes visible information, actions, and lifecycle constraints while preserving native history management.

3.2 TASK AND REAL-TIME INTERACTION

Task and observations. The default task uses randomized 12×12 maps containing Capitals, Cities, and impassable Mountains. Each side explores the map, captures territory, and moves troops to attack the enemy Capital while protecting its own. Capitals and captured Cities generate troops every 2 ticks, and owned plain cells every 50 ticks. Troops move between orthogonally adjacent cells, leaving one unit at the source. The agent may send all movable units or half of the source force. Friendly troops combine, while opposing forces subtract in combat. Capturing the enemy Capital wins the match. The match is drawn if neither side wins before the limit (Appx. C.3).

The engine represents the current view as coordinate-indexed text, describing visible cells by terrain, ownership, and troop counts. Previously explored cells outside current vision retain static terrain, while their current ownership and troop counts are hidden. The observation identifies the team's own Capital and confirmed enemy Capital locations, and marks unexplored regions as unknown. This text representation supports reasoning without visual capability, as detailed in Appx. D.2.

Actions and organization. Actors interact through structured commands that follow a shared action schema, with one action processed per response. Actors can move troops, pass, update a private note, or communicate with teammates. In the multi-agent configuration, the root can also spawn children, assign or reclaim cell control, change their tasks, and terminate them. Children maintain independent contexts and private notes while sharing team vision. They cannot create further children. The default protocol imposes no finite child-session cap, and all actors on a side share a runtime endpoint and its serving capacity. Organization and messaging occupy decision opportunities just as movement does. Figure 3 illustrates how these primitive actions support delegation, force concentration, and capture. Tactics such as rallying are produced through sequences of actions.

Asynchronous execution. The environment advances by one tick per wall-clock second without waiting for model inference. Each actor has at most one decision request in flight, with an observation supplied when issued. Returned commands are checked against the current state and applied at engine ticks. Invalid or stale commands are discarded without execution or automatic replay. The next decision uses a fresh observation while the game continues. Although further reasoning may improve planning, a delayed action can become invalid or miss a tactical opportunity even if legal. Match outcomes thus reflect decision quality and response time (Appx. C.3 and D).

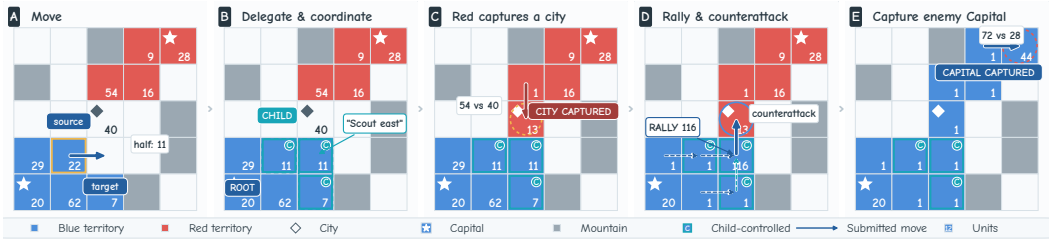


Figure 3: **From actions to tactics.** Schematics of (A) troop movement, (B) child delegation and coordination, and (C–E) City capture, troop rallying, and Capital capture.

3.3 Persistent Competitive Ladder

Within each experiment, all configurations share a ladder. The scheduler considers waiting time, rating-graph connectivity, estimated strength, and match history, relaxing matching bands as waiting increases. It seeks informative comparisons, discourages rematches, and uses available runtime instances. New configurations establish their standing by competing against existing entrants.

Wins, draws, and losses are encoded as 1, 0.5, and 0 and used to update competitive Rating and rating deviation (RD) through Glicko-2 (Glickman, 2022). We process accepted games in recorded order, each as an incremental rating period, updating both participants from their pre-game states. Unrecoverable post-start interruptions classified as participant failures are recorded as technical losses. Failures in evaluation-side lifecycle management or adapter integration invalidate the affected game and are excluded even when only one actor is affected, following the attribution rules in Appx. C.6. Rating therefore summarizes competitive play and participant-attributed failures (Appx. A and C).

Since rating is relative to the entrant pool and realized opponents, we use a separate ladder for each main experiment and compare configurations within it. RD describes uncertainty in the fitted Rating. Robustness checks cover independent random ladders, late entry, and fixed scenarios (Appx. B).

3.4 Availability-Aware Service Evaluation

In routine use, subscription-based agent services must remain accessible as requests accumulate and concurrent demand changes. Usage quotas, rate limits, and concurrency constraints can prevent access even when the underlying configuration performs well. To evaluate this practical dimension alongside competitive performance, GeneralsArena measures service availability through API probes at scheduled match opportunities. ESR combines this availability with Rating to summarize performance and access under the evaluation workload.

At each scheduled pairing, a minimal completion request probes whether each service can start a match. Quota, rate-limit, and unavailability failures count as unsuccessful access. Pairings canceled because the opponent cannot start are excluded for a service whose probe succeeds.

For service i, given the number of accepted starts a_i, the resolved valid opportunities $o_i > 0$, and competitive Rating R_i, we define availability A_i, rating-derived score S_i, and delivered utility U_i as

$$A_i = \frac{a_i}{o_i}, \qquad S_i = \frac{1}{1 + 10^{(1500 - R_i)/400}}, \qquad U_i = A_i S_i. \tag{2}$$

The logistic transform maps competitive Rating R_i to a score using a fixed reference Rating of 1500. This reference is shared within the ladder, with observed win rate reported separately. Multiplying by A_i weights the score by the fraction of delivered opportunities, assigning zero utility to unaccepted opportunities.

ESR expresses the resulting utility on the same numerical scale as Rating,

$$\mathrm{ESR}_i = 1500 + 400 \log_{10} \frac{\widetilde{U}_i}{1 - \widetilde{U}_i}, \qquad \widetilde{U}_i = \mathrm{clip}(U_i, 0.01, 0.99). \tag{3}$$

For fixed competitive Rating, ESR is non-decreasing in availability, with saturation at the clipping bounds. Reporting R_i, RD, A_i, and ESR together describes competitive performance and access under the evaluated service conditions (Appx. C.4).

Table 1: **Runtime-service results.** Configurations are ordered by Rating, and latency follows Eq. (9).

#	Model	Configuration	Rating	RD	WR %	Latency (s)
1	Qwen3.6-35B-A3B	SGL/FP8/off	**1880.2**	63.2	**66.4**	1.67
2	Gemma4-26B-A4B	SGL/BF16/off	1777.2	61.5	44.8	2.64
3	Ornith-1.5-35B-A3B	SGL/FP8/off	1771.7	61.5	47.0	2.53
4	Gemma4-26B-A4B	SGL/BF16/on	1766.3	62.3	42.2	9.03
5	Qwen3.8-27B + DSpark	SGL/FP8/low	1548.1	64.0	36.4	13.52
6	Qwen3.8-27B	SGL/FP8/off	1512.9	63.8	28.6	8.51
7	Qwen3.8-27B	vLLM/FP8/low	1423.3	63.0	24.2	24.34
8	Qwen3.8-27B	SGL/BF16/low	1413.0	62.9	26.1	23.71
9	Qwen3.8-27B	SGL/FP8/low	1409.9	63.1	13.3	33.21
10	Muse-Glimmer-30B	SGL/BF16/low	1391.7	64.0	5.3	13.99
11	Laguna-XS-2.1-33B-A3B	SGL/FP8/on	1376.8	63.0	9.8	8.04
12	Gemma4-26B-A4B	vLLM/BF16/off	1371.1	63.3	9.2	1.14
13	Granite-4.2-30B	SGL/BF16/off	1348.5	64.0	0.0	10.30
14	Qwen3.8-27B	SGL/FP8/xhigh	1343.0	63.0	1.7	56.12
15	Qwen3.8-27B	Trans/FP8/low	1336.6	63.6	1.7	98.42
16	Nemotron3.5-Lightning	SGL/BF16/off	1310.9	63.2	7.8	1.32

Configuration = engine / precision / reasoning. SGL = SGLang. Trans = Transformers. RD = rating deviation. WR = win rate. Latency = request-weighted mean of actor-session medians. DSpark is a speculative decoding method.

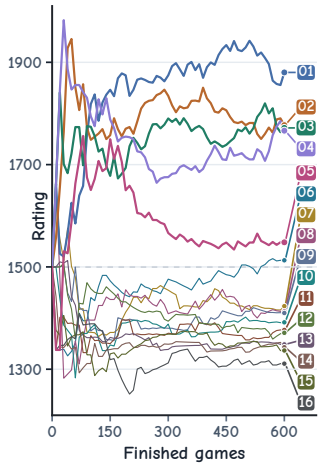


Figure 4: **Rating evolution curves.** IDs refer to Table 1.

4 Experiments

Experimental settings. We evaluate complete entrants defined in Subsec. 3.1 using three harnesses and two fixed strategy prompts. The harnesses retain different approaches to context management and tool execution. Reference uses custom persistent sessions and deterministic compression. Pi (Zechner, 2025) uses native sessions with explicit compaction, and Codex (OpenAI, n.d.) uses persistent application-server threads with automatic compaction. We provide two predefined strategy prompts. Neutral states the objective without prescribing tactics, while Detailed adds tactical and coordination guidance. Their text is shared across models and harnesses. Unless stated otherwise, we use Reference/Neutral in the real-time Default setting, which uses text observations and enables child delegation. Details are in Appx. C and D.

Three complementary experiments use separate adaptive ladders. Subsection 4.1 compares 16 local runtimes over 600 games, varying model, serving engine, precision, and reasoning, with one NVIDIA H20 GPU per instance. Subsection 4.2 crosses these runtimes with three harnesses and two prompts for 96 entrants and 1,920 games. Subsection 4.3 evaluates 24 model–provider–plan configurations over 1,800 games using purchased agent, coding, or token subscriptions from ByteDance Volcengine, GLM, Kimi, MiniMax, Alibaba Bailian, and Xiaomi MiMo. We measure performance and availability through plan-supplied API endpoints with provider-default inference settings under the evaluation workload. Three additional 600-game ladders compare single-actor organization, image observations, and turn-based execution against the Default runtime setting, as presented in Subsec. 4.4. The image ladder contains twelve vision-capable services, and the other comparisons retain all sixteen. Details of the three specific protocols are in Appx. F.

Metrics. We report competitive Rating, RD, and win rate. For local runtimes, latency is the request-weighted mean of actor-session median request times in seconds, as defined in Appx. C.5. Subscription results additionally report availability A and ESR, defined in Eqs. (2) and (3). Full results appear in Appx. E, ordered by Rating for local configurations and ESR for subscription services.

4.1 Runtime-Service Evaluation

Model and runtime configuration jointly determine real-time performance. Qwen3.6-35B-A3B with SGLang, FP8, and reasoning off leads Table 1 at Rating 1880.2 and a 66.4% win rate. Model identity alone does not explain the ordering. Three configurations of Gemma4-26B-A4B obtain Ratings of 1777.2 with SGLang and reasoning off, 1766.3 with SGLang and reasoning on, and 1371.1 with vLLM and reasoning off, spanning 406.0 points despite sharing the same model artifact. Their latency summaries are 2.64, 9.03, and 1.14 seconds, respectively. The SGLang configurations

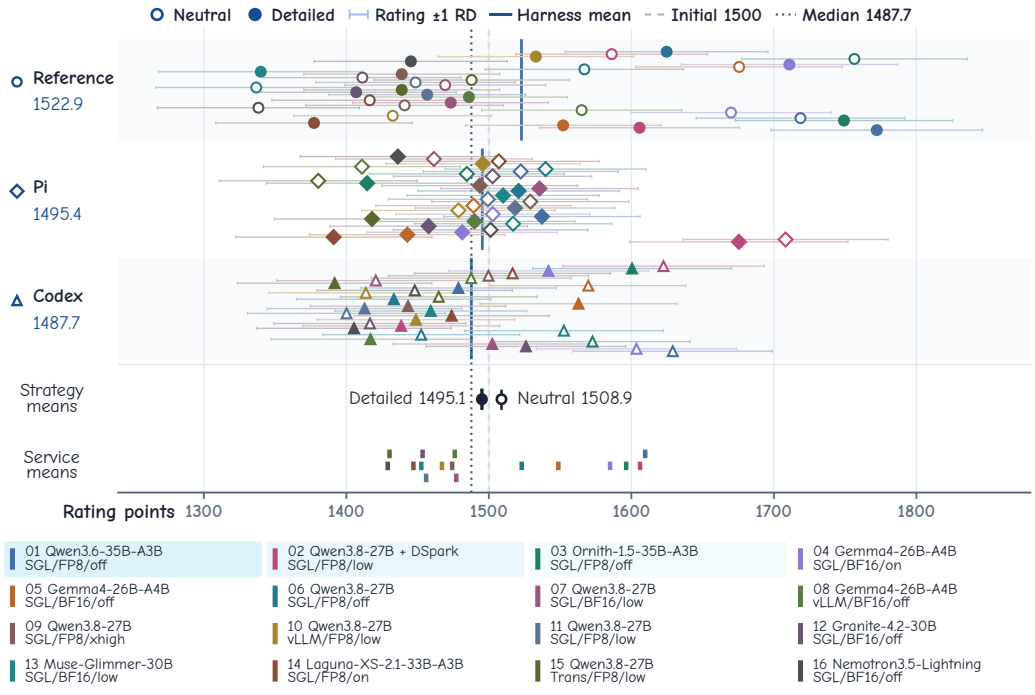


Figure 5: **Runtime service–harness–strategy prompt results.** Whiskers show Rating ± RD for 96 entrants. Colors identify runtime services, shapes identify harnesses, and open/filled markers denote prompts. Solid vertical lines mark each harness's mean Rating. Lower markers show equally weighted entrant means, with the legend ordered by mean Rating.

have similar Ratings despite different reasoning settings and response times. vLLM has the lowest reported latency but ranks below both, showing that low latency alone does not ensure effective play.

Reasoning effort and serving efficiency affect the same outcome. Qwen3.8-27B illustrates the trade-off from another direction. With SGLang and FP8, the off, low, and xhigh reasoning settings yield Ratings of 1512.9, 1409.9, and 1343.0, while the latency summary increases from 8.51 to 33.21 and 56.12 seconds. At low reasoning, the configuration using DSpark (Cheng et al., 2026), a speculative decoding method, reaches 1548.1 with a 13.52-second latency summary. Longer deliberation is therefore not consistently rewarded, while the faster DSpark configuration improves on the other low-reasoning alternatives. These Gemma and Qwen comparisons show how Rating captures the combined effects of serving choices on decision quality and execution efficiency.

Stronger services separate early, with continued movement within groups. Figure 4 shows leading services separating early while similarly performing configurations continue to exchange ranks. Qwen3.6 leads at Rating 1880.2. The top four span 1766.3–1880.2, followed by Qwen3.8 with DSpark at 1548.1 and a dense lower-ranked group (Appx. B and E).

4.2 Runtime Service–Harness–Strategy Prompt Interactions

Performance depends on the complete runtime–harness–prompt configuration. This experiment compares 96 entrants combining 16 runtime services, three harnesses, and two prompts, as shown in Fig. 5. Qwen3.6-35B-A3B with Reference and Detailed has the highest endpoint Rating, 1772.2, compared with 1718.7 for Reference/Neutral, 1537.3 for Pi/Detailed, and 1478.7 for Codex/Detailed. Averaging all six pairings, Qwen3.6 still ranks first. Alongside Qwen3.8/DSpark, Ornith, and the two Gemma4/SGLang configurations, it forms the same leading five-service set as in Subsec. 4.1, although their internal order changes. Thus aggregate service rankings remain similar, while individual outcomes depend on the full configuration.

Harness effects vary across runtime services. Reference has the highest mean Rating at 1522.9, followed by Pi at 1495.4 and Codex at 1487.7. Five of the six highest-rated entrants use Reference. This aggregate pattern nevertheless includes markedly different responses to the harness. Ornith



Seed = Doubao Seed. DS = DeepSeek. Ark Code = Ark Code Latest. MM = MiniMax. Avail. = availability A.

Figure 6: **Subscription-service results.** The 24 Reference/Neutral configurations use provider-default inference over 1,800 games. Rows follow ESR. Circles show Rating, diamonds indicate ESR, and whiskers represent Rating $\pm$ RD. Segments show the availability adjustment.

averages 1752.9 under Reference and 1449.6 under Pi across the two prompts, whereas Qwen3.8 with DSpark averages 1691.9 under Pi and 1596.0 under Reference. Reference's overall lead reflects strong service pairings, not a uniform advantage across the roster.

Prompt effects vary across runtime–harness combinations. Compared with Neutral, Detailed's mean Rating is 5.6 points higher under Reference, 13.5 lower under Pi, and 33.4 lower under Codex. Under Reference, Qwen3.6 scores 1772.2 with Detailed against 1718.7 with Neutral, while Gemma4/SGLang with reasoning off scores 1552.1 against 1675.6. For Qwen3.6, Detailed reverses direction under Codex, scoring 1478.7 against Neutral's 1629.0 (Appx. E.2).

4.3 Subscription-Based Service Evaluation

Availability changes the leading service configuration. Volcengine Doubao Seed 2.0 Lite has the highest ESR at 1675.1, equal to its Rating at full measured availability, as demonstrated in Fig. 6. Bailian Qwen3.7 Max achieves the highest competitive Rating, 1713.7, but its 66.7% availability lowers ESR to 1511.0, placing it ninth. Only three configurations appear in both the Rating and ESR top-five sets. ESR reflects the practical effect of plan-related access constraints by weighting competitive performance by the fraction of valid opportunities the service can accept.

Reliable access matters alongside competitive performance. Eleven configurations have full measured availability, while ten fall below 90%. All nine Volcengine configurations have full measured availability, and four occupy the ESR top five. Their Ratings nevertheless range from 1346.4 to 1675.1, indicating that consistent access alone does not ensure strong play. Bailian Qwen3.8 Max encounters quota rejections, accepting 24 of 40 valid opportunities (60.0% availability). Its Rating of 1281.6 falls to ESR 1174.1, ranking twenty-third. Qwen3.8 Flash on the same provider reaches 89.7% availability and ESR 1557.5 (Appx. C.4).

Services sharing a model label differ across providers. DeepSeek V4 Flash reaches Rating 1697.8 through Bailian versus 1641.4 through Volcengine, but Bailian's 79.7% availability lowers ESR to 1573.3, below Volcengine's 1641.4. Kimi K3 reaches ESR 1529.4 through Kimi versus 1373.5 through Volcengine despite lower Kimi availability. MiniMax M3 reaches ESR 1560.5 through Volcengine versus 1443.4 through MiniMax, both fully available. Thus first-party access does not consistently lead. These results compare model–provider–plan configurations at default inference settings. Shared model labels do not imply identical artifacts or serving stacks (Appx. E.3).

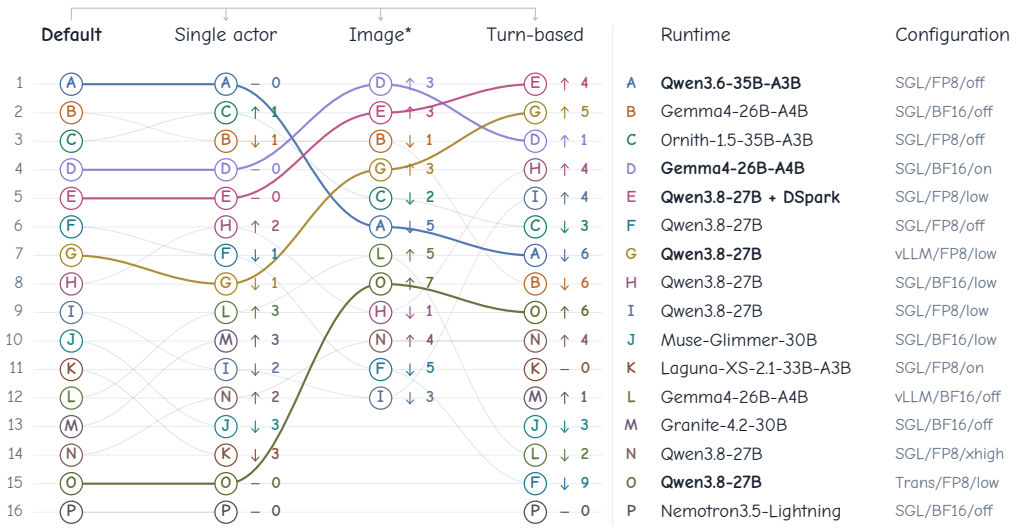


Figure 7: **Runtime rankings across protocols.** Default uses real-time execution, text observations, and multiple actors. Node height shows rank, and arrows show changes. Letters and colors identify configurations. Image* includes twelve vision-capable services. Each ladder is ranked separately and has 600 adaptively matched games with a different opponent schedule.

4.4 Protocol Comparisons

Multi-agent versus single actor. The single-actor ladder retains Default's top-five set in a new order. Qwen3.6 remains first despite using no children in 113 Default games, showing that its lead does not depend on multi-agent organization here. Ornith uses children in 108 of 117 Default games but rises from third to second with one actor. Its wins/draws/losses improve from 55/7/55 to 64/6/37 while latency falls from 2.53 to 1.62 seconds. Ornith exhibits markedly different organizational behavior under multi-agent control, accompanied by substantial performance changes. Performance differences between the single-actor and Default ladders vary across services. Behavior statistics and per-configuration outcomes are reported in Appx. F.

Text versus image observations. Image observations substantially change outcome profiles. Gemma4/SGLang with reasoning on rises from fourth at 49/8/59 in Default to first at 33/63/4 with images. Across the ladder, draws rise from 44.7% in Default to 85.3% (512/600). The image ladder also logs 59,535 rejected decisions among 222,015 completed requests, showing frequent action-validity failures under image input. Qwen3.6 falls from first to sixth and draws 91 of 100 games. In a recorded case in Appx. F.3, it mistakes its own Capital for an enemy target, illustrating the difficulty of interpreting information from raw image inputs.

Real-time versus turn-based execution. Turn-based execution synchronizes actors while game time continues. Qwen3.8/DSpark, Qwen3.8/vLLM with low reasoning, and Gemma4/SGLang with reasoning on rise from fifth, seventh, and fourth to the top three. Qwen3.8/SGLang/FP8 ranks fifth with low reasoning and fifteenth with reasoning off. Gemma4/SGLang ranks third with reasoning on and eighth when off. Several reasoning-enabled configurations rank highly, but Qwen3.8 at xhigh ranks tenth with 58.73-second latency and 69 draws in 71 games. Longer inference delays joint actions and may contribute to time-limit draws.

5 Conclusion

GeneralsArena evaluates complete agent services in persistent real-time competition, linking decision quality to timely action. Across runtimes, harnesses, prompts, providers, and protocols, shared weights do not guarantee shared outcomes, and availability can alter practical rankings. Evaluating the complete service therefore reveals effects hidden by model-only comparisons. Replay analysis shows whether activity becomes task completion. Its persistent ladder also supports reevaluation as services and access conditions evolve. Together, these results support arena-based study of end-to-end agent performance under realistic execution and access constraints.

AI STATEMENT

We used generative AI tools for language polishing and organization, figure preparation, code implementation, and literature searches. We reviewed the code and outputs, checked numerical claims against experiment records, and inspected figures and manuscript content for consistency. We take responsibility for the final text, analysis, code, and artifacts.

REPRODUCIBILITY STATEMENT

Our anonymized project page at `generalsarena.github.io` provides public access to the reported standings, individual match records, and replays. We will release the evaluation code as open source after publication. Appx. A, C, D and H specify the rating procedure, experimental conditions, agent interfaces, and runtime configurations. Appx. B and E to G provide diagnostics, complete results, and qualitative match analyses.

REFERENCES

Alibaba Cloud. Alibaba Cloud Model Studio: qwen3.8-flash. Official model documentation, 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-8-flash.

ByteDance Seed Team. Seed 2.0 Official Launch. Official model release blog, February 2026a. URL https://seed.bytedance.com/en/blog/seed2-0-%E6%AD%A3%E5%BC%8F%E5%8F%91%E5%B8%83.

ByteDance Seed Team. Seed2.1 Officially Released: Advancing AI Productivity. Official model release, June 2026b. URL https://seed.bytedance.com/en/blog/seed2-1-officially-released-advancing-ai-productivity.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A Zaharia, Joseph Gonzalez, and Ion Stoica. Why Do Multi-Agent LLM Systems Fail? In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4082.

Zhekai Chen, Chengqi Duan, Kaiyue Sun, Bohao Li, Yuqing Wang, Manyuan Zhang, and Xihui Liu. UniClawBench: A Universal Benchmark for Proactive Agents on Real-World Tasks. *arXiv preprint arXiv:2607.08768*, 2026.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, Qinyu Chen, Chengqi Deng, Jiping Yu, Damai Dai, Zhengyan Zhang, Yixuan Wei, Yixuan Tan, Wenkai Yang, Runxin Xu, Yu Wu, Zhean Xu, Xuanyu Wang, Muyang Chen, Rui Tian, Xiao Bi, Zhewen Hao, Shaoyuan Chen, Huanqi Cao, Wentao Zhang, Anyi Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. DSpark: Confidence-Scheduled Speculative Decoding with Semi-Autoregressive Generation. *arXiv preprint arXiv:2607.05147*, 2026.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, July 2024.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua M Clymer, and Arjun Yadav. GameBench: Evaluating Strategic Reasoning Abilities of LLM Agents. In *Language Gamification - NeurIPS 2024 Workshop*, 2024.

DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How Capable are Web Agents at Solving Common Knowledge Work Tasks? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 2024.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in Statistics: Methodology and Distribution*, pp. 569–593. Springer, 1992. doi: 10.1007/978-1-4612-4380-9_41.

Gemma Team. Gemma 4 26B A4B IT. Hugging Face model card, 2026. URL https://huggingface.co/google/gemma-4-26B-A4B-it.

Mark E. Glickman. Example of the Glicko-2 System. Technical report, Boston University, 2022. URL https://www.glicko.net/glicko/glicko2.pdf.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. TextArena. *arXiv preprint arXiv:2504.11442*, 2025.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments. Official software framework, Zenodo, 2026. URL https://doi.org/10.5281/zenodo.20953922. Concept DOI.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

IBM Granite Team. Granite-4.2-30B. Hugging Face model card, 2026. URL https://huggingface.co/ibm-granite/granite-4.2-30b.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can Language Models Resolve Real-world Github Issues? In *The Twelfth International Conference on Learning Representations*, 2024.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. AI Agents That Matter. *Transactions on Machine Learning Research*, 2025.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue

Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi K3: Open Frontier Intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast Inference from Transformers via Speculative Decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023.

Keyu Li, Junhao Shi, Yang Xiao, Mohan Jiang, Jie Sun, Yunze Wu, Dayuan Fu, Shijie Xia, Xiaojie Cai, Tianze Xu, Weiye Si, Wenjie Li, Dequan Wang, and Pengfei Liu. AgencyBench: Benchmarking the Frontiers of Autonomous Agents in 1M-Token Real-World Contexts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7422–7440. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.337.

Mingxian Lin, Shengju Qian, Yuqi Liu, Yi-Hua Huang, Yiyu Wang, Wei Huang, Yitang Li, Fan Zhang, Zeyu Hu, Lingting Zhu, Xin Wang, and Xiaojuan Qi. OmniGameArena: A Unified UE5 Benchmark for VLM Game Agents with Improvement Dynamics. *arXiv preprint arXiv:2606.09826*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*, 2024.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74325–74362. Curran Associates, Inc., 2024. doi: 10.52202/079017-2365.

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026.

Meta Superintelligence Labs. Muse Glimmer 30B. Hugging Face model card, 2026. URL https://huggingface.co/meta-models/Muse-Glimmer-30B.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants. In *The Twelfth International Conference on Learning Representations*, 2024.

MiniMax. MiniMax M2.5: Built for Real-World Productivity. Official model release blog, February 2026a. URL https://www.minimax.io/news/minimax-m25.

MiniMax. MiniMax M2.7: Early Echoes of Self-Evolution. Official model release blog, March 2026b. URL https://www.minimax.io/news/minimax-m27-en.

MiniMax. MiniMax M3: Frontier Coding, 1M Context, Native Multimodality — All in One Model. Official model release blog, June 2026c. URL https://www.minimax.io/blog/minimax-m3.

Moonshot AI. Kimi K2.7 Code. Official model release blog, 2026. URL https://www.kimi.ai/resources/kimi-k2-7-code.

NVIDIA. NVIDIA Nemotron 3.5 Lightning 30B A3B. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16.

OpenAI. Codex. Official OpenAI project page, n.d. URL https://openai.com/codex/.

Ornith Team. Ornith-1.5-35B-A3B-FP8. Hugging Face model card, 2026. URL https://huggingface.co/ornith-ai/Ornith-1.5-35B-A3B-FP8.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking Agentic LLM and VLM Reasoning On Games. In *The Thirteenth International Conference on Learning Representations*, 2025.

Poolside. Laguna XS 2.1. Hugging Face model card, 2026. URL https://huggingface.co/poolside/Laguna-XS-2.1-FP8.

Qwen Team. Qwen3.6-35B-A3B-FP8. Hugging Face model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.6-35B-A3B-FP8.

Qwen Team. Qwen3.7: The Agent Frontier. Official model release, May 2026b. URL https://qwen.ai/blog?id=qwen3.7.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Official model release blog, August 2026c. URL https://qwen.ai/blog?id=qwen3.8.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026d. URL https://huggingface.co/Qwen/Qwen3.8-27B.

RadixArk. Qwen3.8-27B-DSpark. Hugging Face model card, 2026. URL https://huggingface.co/RadixArk/Qwen3.8-27B-DSpark.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

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

Volcengine. Doubao-Seed-Evolving. Official model release announcements, 2026. URL https://docs.volcengine.com/docs/ark/model-release-announcement?lang=zh.

Volcengine Service Assistant. Connecting OpenClaw to Volcengine Coding Plan. First-party Coding Plan integration guide, 2026. URL https://developer.volcengine.com/articles/7615528054736945158.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. In *First Conference on Language Modeling*, 2024.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Demystifying LLM-Based Software Engineering Agents. *Proceedings of the ACM on Software Engineering*, 2(FSE): FSE037, 2025. doi: 10.1145/3715754.

Xiaomi MiMo Team. Xiaomi MiMo-V2.5 Series Large Model Launches Public Beta. Official model release blog, 2026. URL https://mimo.mi.com/docs/en-US/news/latest/v2.5-news.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52040–52094. Curran Associates, Inc., 2024. doi: 10.52202/079017-1650.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. HarnessBench: Measuring Harness Effects across Models in Realistic Agent Workflows. *arXiv preprint arXiv:2605.27922*, 2026.

Z.ai. GLM-5.2: Built for Long-Horizon Tasks. Official model release blog, June 2026a. URL https://z.ai/blog/glm-5.2.

Z.ai. GLM-5.3-Flash: Frontier Intelligence, Flash Cost. Official model release blog, August 2026b. URL https://z.ai/blog/glm-5.3-flash.

Z.ai. GLM-5.3: Frontier Coding with Emergent Cyber Capabilities. Official model release blog, August 2026c. URL https://z.ai/blog/glm-5.3.

Mario Zechner. Pi Agent Harness. Open-source software project, 2025. URL https://github.com/earendil-works/pi. MIT license.

Alex L. Zhang, Thomas L. Griffiths, Karthik R. Narasimhan, and Ofir Press. VideoGameBench: Can Vision-Language Models complete popular video games? *arXiv preprint arXiv:2505.18134*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient Execution of Structured Language Model Programs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 62557–62583, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*, 2024.

APPENDIX

A RATING AND UNCERTAINTY

A.1 INITIALIZATION AND INCREMENTAL UPDATES

We implement the Glicko-2 update equations (Glickman, 2022) with the incremental rating periods specified below. Each entrant starts at Rating $R = 1500$, rating deviation $\mathrm{RD} = 350$, and volatility $\sigma = 0.06$. We use $\tau = 0.5$. Each accepted game is one rating period. Records are processed in recorded match order, and both players are updated using their pre-game states. Uninvolved entrants retain their states. A win, draw, or loss contributes score $s = 1$, 0.5, or 0. Technical losses use the same loss score.

For an entrant and its opponent j, use the standard Glicko-2 scale-conversion constant $c = 400/\ln 10 \approx 173.7178$ (Glickman, 2022). It converts the base-10 Rating scale to the natural-logarithm scale used internally by Glicko-2:

$$\mu = \frac{R - 1500}{c}, \qquad \phi = \frac{\mathrm{RD}}{c}, \tag{4}$$

$$g_j = \frac{1}{\sqrt{1 + 3\phi_j^2/\pi^2}}, \qquad E = \frac{1}{1 + \exp[-g_j(\mu - \mu_j)]}, \tag{5}$$

$$v = \frac{1}{g_j^2 E(1 - E)}, \qquad \Delta = v g_j (s - E). \tag{6}$$

After computing the new volatility σ', update

$$\phi_* = \sqrt{\phi^2 + \sigma'^2}, \qquad \phi' = \left(\phi_*^{-2} + v^{-1}\right)^{-1/2}, \tag{7}$$

$$\mu' = \mu + \phi'^2 g_j (s - E), \qquad R' = 1500 + c\mu', \qquad \mathrm{RD}' = c\phi'. \tag{8}$$

The opponent uses score $1 - s$ and the same pair of pre-game states. For a draw, $s - E$ is positive when the expected score is below one half, so avoiding defeat against a stronger entrant can improve Rating.

The volatility update follows the standard Glicko-2 procedure (Glickman, 2022). The initialization, one-game rating periods, and ordered accepted records specify the reporting convention.

A.2 RATING, UNCERTAINTY, AND SERVICE UTILITY

Local tables are ordered by descending Rating. One-RD whiskers indicate uncertainty in the fitted Rating. Subscription-service ESR is calculated from point-estimate Rating and measured availability using Eqs. (2) and (3), with reference Rating 1500 and clipping interval $[0.01, 0.99]$. Segments connecting Rating to ESR show the magnitude of the availability adjustment.

Each ladder is fitted to its own entrant pool and realized opponents, so absolute Rating differences across the three experiments are not directly comparable. Both Rating and ESR summarize performance within a particular ladder and can obscure non-transitive matchup relationships.

The reference 1500 is a reporting convention aligned with Rating initialization, not an empirically selected user-demand profile. The logistic score is a transformation of fitted Rating, not a separately calibrated win probability against a deployed reference agent. Multiplication by availability assigns zero score to an unaccepted counted opportunity under Appx. C.4. ESR therefore summarizes this access convention and reference choice rather than general service quality or expected utility for arbitrary tasks. Changing the reference can change the balance between access and competitive strength, and thus reverse pairwise ESR orderings even with unchanged observations. The displayed RD quantifies Rating uncertainty, not uncertainty in availability or the combined ESR.

B Ladder Robustness and Diagnostic Checks

B.1 Within-Run Rating Evolution

The eventual leading group separates early, while nearby configurations continue to exchange positions. Table 2 tracks all sixteen services through the main runtime ladder and its continuation to 800 games. The main runtime results report 600 games, and the 800-game checkpoint continues the same run. From 200 games onward, the leader and top-four membership remain unchanged. Across the final eight 25-game increments up to 600 games, Spearman rank correlations between consecutive checkpoints range from 0.988 to 1.000. These overlapping checkpoints describe within-run stability, not independent replication.

Table 2: **Complete within-run ranks at selected checkpoints.** All sixteen configurations use the Reference harness and Neutral prompt. Rows follow the main runtime ranking. The gray rule separates its top five. Column headings give completed games. The 800-game checkpoint extends the main 600-game run.

Model	Configuration	100	200	400	600	800
Qwen3.6-35B-A3B	SGLang/FP8/off	1	1	1	1	1
Gemma4-26B-A4B	SGLang/BF16/off	4	2	2	2	2
Ornith-1.5-35B-A3B	SGLang/FP8/off	3	4	3	3	3
Gemma4-26B-A4B	SGLang/BF16/on	2	3	4	4	4
Qwen3.8-27B + DSpark	SGLang/FP8/low	5	5	5	5	5
Qwen3.8-27B	SGLang/FP8/off	7	7	6	6	6
Qwen3.8-27B	vLLM/FP8/low	12	9	8	7	8
Qwen3.8-27B	SGLang/BF16/low	8	6	7	8	7
Qwen3.8-27B	SGLang/FP8/low	11	10	11	9	9
Muse-Glimmer-30B	SGLang/BF16/low	10	12	12	10	11
Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	13	11	10	11	10
Gemma4-26B-A4B	vLLM/BF16/off	6	8	9	12	12
Granite-4.2-30B	SGLang/BF16/off	14	13	14	13	13
Qwen3.8-27B	SGLang/FP8/xhigh	16	14	15	14	14
Qwen3.8-27B	Transformers/FP8/low	15	15	13	15	15
Nemotron3.5-Lightning	SGLang/BF16/off	9	16	16	16	16

B.2 Independent Random Ladders

Three independently initialized 600-game ladders use the same sixteen entrants with different scheduler and map seeds. The baseline is reused from the main experiment. All three identify Qwen3.6 as leader and retain the same leading five services, while their internal ordering changes, as shown in Table 3. Rank correlations are 0.947 for baseline versus Run 1, 0.891 for baseline versus Run 2, and 0.950 between the two additional runs.

B.3 Late Entry

We compare admitting Qwen3.6 from the start with admitting it after fifteen services have played 300 games. The late-entry branch continues those records, then schedules new matches adaptively. Table 4 compares the ranks of all sixteen services after 300, 450, and 600 cumulative ladder games. At 300 games, the delayed branch includes Qwen3.6 immediately after admission at its initial Rating of 1500, before it has played a match. By 450 games, it leads both branches. Following late admission, it reaches fourth after ten of its own games and first after twenty, and finishes first at Rating 1885.7 after 65 games. The from-start baseline finishes first at 1880.2 after 113 games. These results demonstrate that a strong agent system can quickly reach a leading position even after late admission, with rating-based adaptive matchmaking connecting it to established opponents.

Table 3: **Complete results across independent ladders.** All sixteen configurations use the Reference harness and Neutral prompt. Rows follow the main runtime ranking. The gray rule separates its top five. Cells show rank / Rating. Baseline is the main runtime experiment.

Model	Configuration	Baseline	Run 1	Run 2
Qwen3.6-35B-A3B	SGLang/FP8/off	1 / 1880.2	1 / 1831.7	1 / 1846.6
Gemma4-26B-A4B	SGLang/BF16/off	2 / 1777.2	4 / 1684.5	4 / 1690.9
Ornith-1.5-35B-A3B	SGLang/FP8/off	3 / 1771.7	2 / 1779.5	3 / 1693.6
Gemma4-26B-A4B	SGLang/BF16/on	4 / 1766.3	3 / 1694.7	2 / 1797.8
Qwen3.8-27B + DSpark	SGLang/FP8/low	5 / 1548.1	5 / 1613.1	5 / 1636.9
Qwen3.8-27B	SGLang/FP8/off	6 / 1512.9	6 / 1466.1	6 / 1506.7
Qwen3.8-27B	vLLM/FP8/low	7 / 1423.3	7 / 1457.8	8 / 1429.1
Qwen3.8-27B	SGLang/BF16/low	8 / 1413.0	8 / 1442.8	7 / 1472.9
Qwen3.8-27B	SGLang/FP8/low	9 / 1409.9	10 / 1387.5	12 / 1358.8
Muse-Glimmer-30B	SGLang/BF16/low	10 / 1391.7	14 / 1280.5	14 / 1339.1
Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	11 / 1376.8	11 / 1351.6	13 / 1348.0
Gemma4-26B-A4B	vLLM/BF16/off	12 / 1371.1	9 / 1424.3	11 / 1370.6
Granite-4.2-30B	SGLang/BF16/off	13 / 1348.5	12 / 1342.5	9 / 1394.8
Qwen3.8-27B	SGLang/FP8/xhigh	14 / 1343.0	13 / 1327.8	10 / 1370.9
Qwen3.8-27B	Transformers/FP8/low	15 / 1336.6	16 / 1266.3	16 / 1305.1
Nemotron3.5-Lightning	SGLang/BF16/off	16 / 1310.9	15 / 1272.7	15 / 1311.2

Table 4: **Complete rank comparison with late entry.** All sixteen configurations use the Reference harness and Neutral prompt. Rows follow the main runtime ranking. The gray rule separates its top five. Each group gives cumulative ladder games, comparing admission from the start with Qwen3.6 admission after 300 games. The 300-game late-entry column includes Qwen3.6 immediately after admission at its initial Rating of 1500, before its first match.

		300 games		450 games		600 games	
Model	Configuration	From start	Late entry	From start	Late entry	From start	Late entry
Qwen3.6-35B-A3B	SGLang/FP8/off	1	7	1	1	1	1
Gemma4-26B-A4B	SGLang/BF16/off	2	2	2	3	2	4
Ornith-1.5-35B-A3B	SGLang/FP8/off	3	1	3	2	3	2
Gemma4-26B-A4B	SGLang/BF16/on	4	3	4	4	4	3
Qwen3.8-27B + DSpark	SGLang/FP8/low	5	5	5	5	5	5
Qwen3.8-27B	SGLang/FP8/off	6	6	6	6	6	6
Qwen3.8-27B	vLLM/FP8/low	7	8	8	7	7	7
Qwen3.8-27B	SGLang/BF16/low	8	4	7	8	8	8
Qwen3.8-27B	SGLang/FP8/low	11	9	9	9	9	9
Muse-Glimmer-30B	SGLang/BF16/low	10	12	10	13	10	13
Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	12	13	12	12	11	12
Gemma4-26B-A4B	vLLM/BF16/off	9	10	11	10	12	10
Granite-4.2-30B	SGLang/BF16/off	15	14	14	15	13	14
Qwen3.8-27B	SGLang/FP8/xhigh	14	11	15	11	14	11
Qwen3.8-27B	Transformers/FP8/low	13	15	13	16	15	16
Nemotron3.5-Lightning	SGLang/BF16/off	16	16	16	14	16	15

B.4 FIXED-SCENARIO DIAGNOSTICS

We run ten repetitions of each of ten fixed pair–seed scenarios comprising five near-peer pairs and five wider-gap pairs, defined using the main runtime Ratings. The higher-rated sides record 11 wins, 38 draws, and one loss in near-peer scenarios, versus 33 wins, seventeen draws, and no losses in wider-gap scenarios. Table 5 shows each pair's rank comparison in the main runtime setting alongside its Rating gap and match outcomes. The larger-gap group thus performs better in these selected scenarios. Near-peer scenarios were selected retrospectively from a pilot, and each scenario fixes map and seat orientation. This is a diagnostic comparison rather than prospective validation.

Table 5: **Repeated fixed-scenario diagnostics.** All rows use the Reference harness and Neutral prompt. Each service pair lists the higher-rated configuration above the lower-rated one. Rank comparison gives the corresponding ranks in Table 1. Rating gaps and W/D/L are expressed from the higher-rated side.

Group	Service pair (model / configuration)	Rank comparison	Seed	Rating gap	W/D/L
Near-peer	Gemma4-26B-A4B SGLang/BF16/off Gemma4-26B-A4B SGLang/BF16/on	2 vs 4	160003	10.9	4/6/0
Near-peer	Granite-4.2-30B SGLang/BF16/off Nemotron3.5-Lightning SGLang/BF16/off	13 vs 16	160009	37.6	4/6/0
Near-peer	Muse-Glimmer-30B SGLang/BF16/low Laguna-XS-2.1-33B-A3B SGLang/FP8/on	10 vs 11	160008	14.9	0/9/1
Near-peer	Qwen3.8-27B vLLM/FP8/low Qwen3.8-27B SGLang/FP8/low	7 vs 9	160001	13.4	1/9/0
Near-peer	Qwen3.8-27B SGLang/BF16/low Qwen3.8-27B Transformers/FP8/low	8 vs 15	160006	76.4	2/8/0
Wider-gap	Qwen3.6-35B-A3B SGLang/FP8/off Nemotron3.5-Lightning SGLang/BF16/off	1 vs 16	171000	569.3	10/0/0
Wider-gap	Gemma4-26B-A4B SGLang/BF16/off Qwen3.8-27B Transformers/FP8/low	2 vs 15	171001	440.6	3/7/0
Wider-gap	Ornith-1.5-35B-A3B SGLang/FP8/off Granite-4.2-30B SGLang/BF16/off	3 vs 13	171002	423.2	3/7/0
Wider-gap	Gemma4-26B-A4B SGLang/BF16/on Qwen3.8-27B SGLang/FP8/xhigh	4 vs 14	171003	423.3	10/0/0
Wider-gap	Qwen3.6-35B-A3B SGLang/FP8/off Qwen3.8-27B Transformers/FP8/low	1 vs 15	171004	543.6	7/3/0

C Experimental Conditions and Service Accounting

C.1 Adaptive Scheduling

Each entrant participates in at most one active game. Pairing favors connected rating comparisons, nearby estimated strengths, and less frequently repeated matchups, with matching bands relaxed as waiting increases. Runtime allocation also affects which entrants are eligible. Thus the realized opponents and number of games per entrant need not be equal, and the main ladders are not paired round-robin experiments.

The main ladders use scheduler seed 8,202 and map-seed base 60,000. The runtime sample contains 57–117 games per entrant, the factorial sample 39–47, and the subscription sample 11–267. These sampling differences are retained in the full results rather than normalized away.

C.2 Local Execution Resources

Each local runtime instance occupies one NVIDIA H20 GPU with tensor parallelism one. A side's root and child sessions share that assigned instance and its serving capacity. The instance is not shared with another active game. The runtime experiment keeps sixteen services resident and supports eight simultaneous games.

The factorial experiment uses the same per-side resource allocation. Each active side receives a separate runtime instance, including when both sides use the same configuration. Loading and warm-up occur before the match clock starts.

The local serving engines are SGLang (Zheng et al., 2024), vLLM (Kwon et al., 2023), and Transformers (Wolf et al., 2020). DSpark uses speculative decoding (Leviathan et al., 2023) through SGLang with a separate BF16 draft artifact. Provider-side resources are not controlled in the subscription study. Those results compare the purchased service configurations under the specified workload. Recorded artifact revisions, sampling profiles, serving versions, and API aliases are provided in Appx. H.

C.3 Maps, Seats, and Termination

Games use asymmetric 12×12 maps with 26–37 Mountains and four neutral Cities containing 40–50 units each. Passable cells are connected. Capitals are at least two cells from the edge and nine Manhattan steps apart, with clear 3×3 neighborhoods. Their reachable-cell counts within seven steps differ by at most five. Seats are assigned pseudorandomly, and the main ladders do not enforce paired seat-swapped matches or equal seat counts.

The engine advances once per second. Capitals and owned Cities produce one unit every two ticks, and owned plain cells produce one every 50 ticks. Moves are orthogonal, retain one source unit, and send either all movable units or half the source force. Friendly arrivals reinforce, while opposing forces subtract. Capturing the opponent's Capital wins. Decisions are validated against the state at execution, so a delayed response may become invalid even if it was legal when requested.

The base match limit is twenty minutes, with activity-dependent extensions up to a hard maximum of 28 minutes. Matches that reach the applicable time limit without a Capital capture are draws. Participant-attributed interruptions are handled as technical losses.

C.4 Subscription Workload and Availability

Evaluation workload. The subscription-service experiments were completed before 21 September 2026 UTC. The reported results reflect the plan terms and access conditions during the evaluation period. Subsequent changes to quota windows, credit charging rules, or service configurations may alter measured availability and ESR in repeated evaluations.

We evaluate endpoints supplied through Volcengine Ark Agent Plan, GLM Coding Plan, Kimi Coding Plan, MiniMax Token Plan, Bailian Token Plan, and Xiaomi MiMo Token Plan. We randomly select 24 newly registered accounts, one per entrant, and activate monthly subscriptions on the same date through each provider's official subscription channel. During evaluation, the plans were subject

to provider-specific limits such as five-hour and weekly usage allowances. The workload permits up to twelve simultaneous games and one active game per entrant. Root and child requests share their entrant's endpoint. Services retain provider-default temperature, top-p, and reasoning settings. The harness output allowance is specified separately. Shared provider infrastructure may still correlate access across configurations despite their separate accounts.

Availability accounting. Availability is an access measurement, separate from the win/draw/loss score used in Rating. Each pre-match probe allows at most three attempts in total. These attempts belong to one scheduled opportunity and do not separately enter the availability denominator or failure count. Availability records the final access outcome rather than individual attempt outcomes. For each *resolved valid opportunity* in the reported sample, an accepted start contributes one and the service's own quota, rate-limit, or unavailability failure contributes zero. Availability is the mean of these binary outcomes. A subsequent game loss, including a technical loss, does not retroactively turn an accepted start into an availability failure.

Outcome for the service	Availability value	Included in denominator
Probe succeeds and the match starts	1	Yes
Its own probe ultimately fails	0	Yes
Probe succeeds, but the opponent fails and the pairing is canceled	Excluded	No

A successful probe for a canceled pairing is excluded from both numerator and denominator.

The frozen ledger contains 3,600 accepted starts, 160 own-service failures (145 quota, twelve rate-limit, and three unavailability failures), and 146 successful probes canceled because the opponent could not start. Thus 3,760 resolved opportunities enter the reported availability values.

Interpretation. Availability measures acceptance among resolved valid opportunities under this workload. Because successful canceled probes are excluded, it depends on realized pairings and the cancellation rule. Ultimately failed probes trigger a cooldown of five minutes, or five hours for quota failures, during which no new offers enter the denominator. Consequently, equal availability values can conceal different periods without access. A value of one means that all counted opportunities were accepted, not that the service was continuously accessible. This measurement is neither standalone probe success nor coverage of a fixed external request stream.

These observations characterize the evaluated service configurations under the joint workload rather than isolated endpoint capacity. We do not estimate recovery time or the fraction of wall-clock demand that would be served. Rating, RD, and availability remain separately reported, while ESR provides a summary under this scheduling policy and the reference convention in Appx. A. Changes to demand, cooldown, or pairing can change availability and the resulting ESR ordering.

Plan selection and usage. We select monthly plans in a comparable price band across providers, with original prices of CNY 99–200. Table 6 lists the plan types, prices, endpoint composition, and mean recorded input and output tokens per completed request.

Table 6: **Monthly subscription plans and recorded token usage.** Prices are undiscounted CNY per month. Avg. input/output are recorded token totals divided by completed requests, pooled by provider. Games without actor metrics are excluded.

Provider	Plan	CNY/month	Endpoints	Avg. input	Avg. output
Volcengine	Ark Agent Plan	200.00	9	8,907	1,796
GLM	Coding Plan	118.00	2	6,824	2,299
Kimi	Coding Plan	199.00	2	5,269	812
MiniMax	Token Plan	119.00	3	10,892	1,197
Bailian	Token Plan	180.00	6	9,296	1,518
MiMo	Token Plan	99.00	2	9,622	994

C.5 Measurement Definitions

For actor-session j, let n_j be its completed-request count and m_j its logged median request latency in milliseconds. The reported latency in seconds is

$$L = \frac{\sum_{j:m_j \text{ recorded}} n_j m_j}{1000 \sum_{j:m_j \text{ recorded}} n_j}. \tag{9}$$

This request-weighted mean excludes sessions without recorded latency. Reference measures model-client completion time, whereas Pi and Codex measure native execution steps, including context-management work. All exclude the subsequent wait for an engine tick. Decision-rejection rates count recorded validation failures per completed request. Their response coverage may differ from the latency statistic. Processed decisions include non-movement actions, technical losses are included in losses, and token totals cover accepted-game actor responses. Marginal Rating means weight entrants equally.

C.6 Failure Attribution

Invalid or stale decisions consume time without executing an action, while play continues. Unrecoverable participant-attributed interruptions after a match starts count as technical losses. Evaluator execution or integration failures invalidate the affected game and are excluded, including failures affecting only one actor. Pre-start access failures enter availability separately, as described in Appx. C.4.

Generation or parsing failures and execution-time rejections are different stages. The runtime-service Reference path can record an invalid decision in the action-processing log, whereas other harness paths can reject a decision before it becomes an execution action. Reported counters therefore follow their recorded stage rather than treating all invalid outputs as the same event. The case-level reconciliation in Table 18 distinguishes these counts.

D AGENT HARNESSES AND EXPERIMENTAL PROMPTS

D.1 SHARED INTERFACE AND HARNESS CONFIGURATION

A shared adapter connects Reference, Pi, and Codex to the same observation, action, and lifecycle interface. Each root or child actor has its own persistent session, private notes, and at most one request in flight. The latest observation is supplied as transient input. Reference assembles rules, the frozen strategy prompt, notes, recent actions, and a deterministic checkpoint, compressing at 46.875% of the configured context window while normally retaining sixteen recent turns. Pi uses native sessions with explicit compaction, reserving 34,816 tokens for compaction and retaining 8,192 recent tokens. Codex uses persistent application-server threads with native automatic compaction and a configured 30,720-token compaction limit. Both native harnesses disable parallel tool calls and require a tool response for action requests. All three configure a 32,768-token output allowance. Pi and Codex cap this allowance by the remaining context budget, while Reference further compresses history when needed to reserve it. This allowance is not a verified model-side hard limit. The runtime supplies the context limit and inference settings. Context-management and tool-execution time remain part of the actor's execution time.

D.2 OBSERVATION REPRESENTATION

We divide the map into three parts: currently visible cells, previously explored cells outside current vision, and unexplored cells. Visible cells expose terrain, ownership, and troop counts. Previously explored cells retain their terrain while current ownership and troop counts remain unknown. Unexplored cells are marked as unknown. Team vision is shared by all actors. The observation marks the team's own Capital and retains the locations of enemy Capitals once discovered. Text and image observations represent these three visibility states in their respective formats.

D.3 ACTION TOOL INTERFACE AND FEEDBACK

The adapters supply a frozen action schema through their native tool interfaces and attach the current observation to each decision request. The schema defines an action selector and its associated parameters: a movement target and troop amount, note text, message recipient and content, or child-task and cell-control assignments. The interface processes one structured action per decision. Actors can move troops, pass, update private notes, and send messages. Roots can additionally create children, assign or reclaim cell control, update tasks, and terminate children. Children cannot create further children. Malformed or role-incompatible actions are rejected. Reference requires exactly one tool call. Pi and Codex retain the first when multiple calls are returned. Their native schemas need not be identical, but returned actions pass through shared action and role checks. Actions are checked again against the current state when executed, and rejected decisions return feedback while the game continues. Unrecoverable interruptions follow Appx. C.6.

D.4 FROZEN STRATEGY PROMPT TEXT

The same Neutral and Detailed strategy prompts are used across services and harnesses without model-specific tuning. Their text is reproduced below with line wrapping for readability.

Neutral.

```
Pursue the stated objective using the current observation. Choose
organization and movement actions naturally without following a
prescribed tactical policy.
```

Detailed.

```
Use a deliberate general strategy:
- Maintain a concise current objective and update persistent notes
  when plans change.
- Prefer useful non-friendly captures that expand vision or improve
  a viable front; avoid repeated friendly transfers that do not
  advance a concrete plan.
```

```
- Balance expansion with headquarters defense, consolidate enough
  units before attacking defended cells, and redirect forces when
  a route stalls.
- Root agents should use child agents only for explicit independent
  responsibilities. Assign controlled cells and a concise task,
  coordinate through short messages, and reclaim or redirect
  resources when a child is idle or blocked.
- Exploit independent actor pipelines on separate fronts instead of
  duplicating the same work.
- Once an enemy headquarters is revealed, preserve required
  defenses and concentrate sufficient force along a legal route
  to capture it.
```

Common game-rule prompt. The shared game-rule portion of the prompt is reproduced below as an excerpt, whereas the strategy prompts above are reproduced in full. The prompts use "headquarters" for the cells called Capitals in the paper.

```
You are an agent controlling one actor in a real-time strategy game.

Objective: win by capturing every enemy headquarters. If multiple players
survive the time limit, the result is a tie.

Rules:
- The map contains plain cells, cities, impassable
  mountains, and headquarters.
- GAME gives map width and height, the current tick, the public max_ticks
  deadline, and ticks_remaining.
- ACTOR_STATE and TEAM_STATE are exact summaries of the
  visible team-owned state.
- Fog reveals the cells owned by your team and their immediate neighbors.
- OWN_HEADQUARTERS is your team's own starting headquarters, never
  an enemy target.
- VISIBLE_CELLS is a compact table. Each JSON array follows the preceding
  COLUMNS declaration and gives exact row, column, terrain, owner class,
  units, and the controller for friendly cells. Owner classes are none,
  neutral, friendly, and enemy_N. A headquarters with owner friendly is
  your own team's headquarters.
- REMEMBERED_TERRAIN is a compact table containing only terrain for cells
  no longer visible. Ownership and units there are unknown.
- UNEXPLORED_RANGES is the exact complement of explored terrain within
  the public map bounds. Each row gives an inclusive column range that
  has never been seen.
- REVEALED_ENEMY_HQS is the only authoritative source for enemy
  headquarters locations. If it is {}, no enemy headquarters location is
  known. A city or an ordinary enemy cell is never evidence that it
  is a headquarters.
- ELIMINATED_PLAYERS lists players already removed by headquarters
  capture. Their former headquarters are no longer objectives and are
  omitted from REVEALED_ENEMY_HQS.
- Moving between friendly cells only transfers units. It does not capture
  territory or reveal new cells. Capturing a non-friendly cell
  may expand vision.
- Headquarters and cities produce units. Owned plain cells produce
  units more slowly.
- A move sends all units except one, or half rounded down, to
  one orthogonal neighbor.
- Combat subtracts opposing forces. Capturing an enemy
  headquarters eliminates them.
- A contacted enemy headquarters remains revealed.
- Every active actor has an independent decision pipeline.
- Every decision may issue exactly one action. Organization actions
  consume the same action slot as movement.
- set_note replaces a persistent private note that is included in every
  later decision context for that actor. Repeating the same note has
  no additional effect.
- Each observation is a real-time snapshot. The game can advance while
```

```
  inference is running, so the engine still validates every submitted
  action when it arrives.
- Root agents may create direct child agents and assign cells to them.
- Child agents cannot create child agents.
- Agents on the same team receive the same legal team vision but have
  independent contexts and memories.
- TEAM_ACTORS lists exact actor IDs. send_message.to must select one of
  those IDs; labels such as root are not addresses.

Use only the provided action schema. Do not assume
hidden map information.
```

E Full Results and Supplementary Analysis

Reported Rating differences are computed from unrounded values before rounding for display, so they can differ from differences between the rounded table entries.

E.1 Local Runtime Services

Table 7 reports all sixteen configurations. Beyond endpoint rank, their outcome and execution profiles distinguish winning, avoiding defeat, and producing valid actions. Granite-4.2 wins none of its 57 games but draws 49 and loses eight. Nemotron3.5-Lightning wins five, draws 39, and loses twenty of 64. Granite nevertheless has the higher Rating, illustrating that win fraction alone omits draws and opponent strength.

Gemma4/vLLM with reasoning off has a shorter latency summary than its SGLang counterpart, 1.14 versus 2.64 seconds, but records 7,505 decision rejections among 24,261 completed requests, compared with zero among 20,168 for SGLang. The 30.9% rejection fraction shows why fast responses need not translate into useful actions. In one recorded match, missing `move.send` fields cause repeated validation failures. This example supports the distinction between response speed and executable decisions without attributing the full Rating gap to one error type.

Table 7: **Complete local runtime-service results.** All rows use the Reference harness and Neutral prompt. Configuration gives engine / precision / reasoning. W/D/L denotes wins/draws/losses. All technical-loss counts are zero. Latency follows Eq. (9).

Rank	Model	Configuration	Games	W/D/L	Rating	RD	WR %	Latency (s)
1	Qwen3.6-35B-A3B	SGLang/FP8/off	113	75/8/30	1880.2	63.2	66.4	1.67
2	Gemma4-26B-A4B	SGLang/BF16/off	96	43/16/37	1777.2	61.5	44.8	2.64
3	Ornith-1.5-35B-A3B	SGLang/FP8/off	117	55/7/55	1771.7	61.5	47.0	2.53
4	Gemma4-26B-A4B	SGLang/BF16/on	116	49/8/59	1766.3	62.3	42.2	9.03
5	Qwen3.8-27B + DSpark	SGLang/FP8/low	77	28/22/27	1548.1	64.0	36.4	13.52
6	Qwen3.8-27B	SGLang/FP8/off	63	18/39/6	1512.9	63.8	28.6	8.51
7	Qwen3.8-27B	vLLM/FP8/low	66	16/39/11	1423.3	63.0	24.2	24.34
8	Qwen3.8-27B	SGLang/BF16/low	69	18/29/22	1413.0	62.9	26.1	23.71
9	Qwen3.8-27B	SGLang/FP8/low	60	8/46/6	1409.9	63.1	13.3	33.21
10	Muse-Glimmer-30B	SGLang/BF16/low	57	3/46/8	1391.7	64.0	5.3	13.99
11	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	61	6/45/10	1376.8	63.0	9.8	8.04
12	Gemma4-26B-A4B	vLLM/BF16/off	65	6/46/13	1371.1	63.3	9.2	1.14
13	Granite-4.2-30B	SGLang/BF16/off	57	0/49/8	1348.5	64.0	0.0	10.30
14	Qwen3.8-27B	SGLang/FP8/xhigh	60	1/51/8	1343.0	63.0	1.7	56.12
15	Qwen3.8-27B	Transformers/FP8/low	59	1/46/12	1336.6	63.6	1.7	98.42
16	Nemotron3.5-Lightning	SGLang/BF16/off	64	5/39/20	1310.9	63.2	7.8	1.32

E.2 Runtime Service–Harness–Strategy Prompt Combinations

Tables 8 and 9 cover the complete $16 \times 3 \times 2$ grid of runtime services, harnesses, and strategy prompts. Across the 48 runtime–harness pairs, Detailed raises Rating in twenty and lowers it in 28, with a mean change of -13.8 points. Averaging the two prompts within each runtime, Reference is best for seven services, Pi for eight, and Codex for one. Thus the harness with the highest overall mean need not be preferred by most individual services.

Ornith's harness means range from 1449.6 with Pi to 1752.9 with Reference, a 303.3-point spread. Qwen3.6's Detailed-minus-Neutral change is $+53.6$ with Reference, $+14.9$ with Pi, and -150.4 with Codex. These sign changes expose interactions that a single prompt average conceals.

Descriptive decomposition. For fitted Rating r_{ijk} of runtime i, harness j, and prompt k, we decompose deviations from the grand mean into marginal and interaction terms:

$$r_{ijk} = \bar{r} + a_i + b_j + c_k + (ab)_{ij} + (ac)_{ik} + (bc)_{jk} + (abc)_{ijk}. \tag{10}$$

Each term is centered over its indices. Squared terms yield the shares in Table 10. Runtime accounts for 45.85% and runtime–harness interaction for 39.32%, whereas the prompt main effect accounts for 0.54%. These arithmetic shares describe one fitted ladder, rather than causal effects or independent variance estimates.

Uncertainty. We use 1,024 bootstrap replicates, each drawing 1,920 accepted games independently with replacement (Efron, 1992). Each replicate restarts the 96-entrant ladder from common initialization and updates Rating sequentially in draw order. Repeated games are processed again at their sampled positions rather than restored to the original sequence. Pairwise ordering probabilities count a higher Rating as one and an equal Rating as one half. A dominance edge is added when this probability is at least 0.975. Strongly connected components are collapsed before confidence tiers are constructed. The reported dominance graph is acyclic, so its highest tier contains the nine entrants with no dominator. This limits evidence for a uniquely strongest configuration. Resampling conditions on the realized matches and outcomes and does not regenerate adaptive matchmaking. Separate independent-run checks are provided for the runtime experiment in Appx. B.

The sampling pool follows the recorded match sequence. A single serial Python `random.Random` stream uses seed 201087882, obtained from $8202 + 1920 \times 104729$, across all replicates. Each replicate initializes Rating 1500, RD 350, and volatility 0.06, with $\tau = 0.5$, numerical tolerance 10^{-6}, and scale constant 173.7178. Both participants update from their pre-game states, while other entrants remain unchanged. Entrants absent from a replicate retain their initialization and remain in the comparisons. The saved report contains marginal summaries and aggregate pairwise comparisons, not the full joint replicate matrix.

Table 8: **Runtime service–harness–strategy prompt results, ranks 1–48.** Configuration gives engine / precision / reasoning. W/D/L denotes wins/draws/losses. All technical-loss counts are zero. Latency (s) follows Eq. (9).

Rank	Model	Configuration	Harness	Prompt	Games	W/D/L	Rating	RD	WR %	Latency (s)
1	Qwen3.6-35B-A3B	SGLang/FP8/off	Reference	Detailed	39	22/14/3	1772.2	74.2	56.4	1.55
2	Ornith-1.5-35B-A3B	SGLang/FP8/off	Reference	Neutral	39	23/12/4	1756.5	79.2	59.0	2.52
3	Ornith-1.5-35B-A3B	SGLang/FP8/off	Reference	Detailed	39	22/11/6	1749.2	76.3	56.4	2.44
4	Qwen3.6-35B-A3B	SGLang/FP8/off	Reference	Neutral	39	18/18/3	1718.7	73.1	46.2	1.53
5	Gemma4-26B-A4B	SGLang/BF16/on	Reference	Detailed	39	18/17/4	1710.9	76.0	46.2	10.34
6	Qwen3.8-27B + DSpark	SGLang/FP8/low	Pi	Neutral	39	18/15/6	1708.2	71.9	46.2	11.06
7	Gemma4-26B-A4B	SGLang/BF16/off	Reference	Neutral	39	14/21/4	1675.6	72.5	35.9	2.61
8	Qwen3.8-27B + DSpark	SGLang/FP8/low	Pi	Detailed	39	17/18/4	1675.5	76.5	43.6	12.25
9	Gemma4-26B-A4B	SGLang/BF16/on	Reference	Neutral	40	20/10/10	1669.9	70.4	50.0	9.36
10	Qwen3.6-35B-A3B	SGLang/FP8/off	Codex	Neutral	39	8/29/2	1629.0	70.0	20.5	2.55
11	Qwen3.8-27B	SGLang/FP8/off	Reference	Detailed	39	11/24/4	1624.8	71.2	28.2	9.87
12	Qwen3.8-27B + DSpark	SGLang/FP8/low	Codex	Neutral	40	15/17/8	1622.5	70.6	37.5	19.02
13	Qwen3.8-27B + DSpark	SGLang/FP8/low	Reference	Detailed	43	15/21/7	1605.8	70.2	34.9	15.26
14	Gemma4-26B-A4B	SGLang/BF16/on	Codex	Neutral	39	12/17/10	1603.6	70.2	30.8	16.59
15	Ornith-1.5-35B-A3B	SGLang/FP8/off	Codex	Detailed	39	11/21/7	1600.5	69.7	28.2	3.36
16	Qwen3.8-27B + DSpark	SGLang/FP8/low	Reference	Neutral	47	20/16/11	1586.2	67.2	42.6	13.60
17	Ornith-1.5-35B-A3B	SGLang/FP8/off	Codex	Neutral	42	6/33/3	1572.7	68.4	14.3	3.77
18	Gemma4-26B-A4B	SGLang/BF16/off	Codex	Neutral	41	7/30/4	1569.8	68.4	17.1	3.22
19	Qwen3.8-27B	SGLang/FP8/off	Reference	Neutral	39	8/25/6	1567.0	69.5	20.5	8.24
20	Gemma4-26B-A4B	vLLM/BF16/off	Reference	Neutral	40	4/33/3	1565.1	70.2	10.0	1.00
21	Gemma4-26B-A4B	SGLang/BF16/off	Codex	Detailed	40	5/33/2	1562.9	69.0	12.5	3.18
22	Muse-Glimmer-30B	SGLang/BF16/low	Codex	Neutral	40	9/21/10	1552.6	69.7	22.5	19.36
23	Gemma4-26B-A4B	SGLang/BF16/off	Reference	Detailed	41	8/27/6	1552.1	68.6	19.5	2.85
24	Gemma4-26B-A4B	SGLang/BF16/on	Codex	Detailed	40	7/25/8	1541.8	70.2	17.5	18.00
25	Qwen3.8-27B	SGLang/FP8/off	Pi	Neutral	39	4/31/4	1539.8	70.4	10.3	5.28
26	Qwen3.6-35B-A3B	SGLang/FP8/off	Pi	Detailed	40	1/39/0	1537.3	68.8	2.5	0.64
27	Qwen3.8-27B	SGLang/BF16/low	Pi	Detailed	39	6/25/8	1535.5	69.2	15.4	24.43
28	Qwen3.8-27B	vLLM/FP8/low	Reference	Detailed	40	5/33/2	1533.0	68.8	12.5	27.40
29	Qwen3.8-27B	SGLang/FP8/xhigh	Pi	Neutral	39	2/33/4	1529.1	69.1	5.1	48.04
30	Granite-4.2-30B	SGLang/BF16/off	Codex	Detailed	39	0/34/5	1526.0	70.1	0.0	14.25
31	Qwen3.6-35B-A3B	SGLang/FP8/off	Pi	Neutral	41	0/41/0	1522.4	68.3	0.0	0.75
32	Qwen3.8-27B	SGLang/FP8/off	Pi	Detailed	39	5/30/4	1520.9	70.6	12.8	5.80
33	Qwen3.8-27B	SGLang/FP8/low	Pi	Detailed	39	1/35/3	1518.4	70.0	2.6	33.99
34	Muse-Glimmer-30B	SGLang/BF16/low	Pi	Neutral	40	7/24/9	1517.0	69.2	17.5	15.41
35	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Codex	Neutral	41	4/32/5	1516.7	68.8	9.8	3.96
36	Muse-Glimmer-30B	SGLang/BF16/low	Pi	Detailed	44	5/33/6	1510.0	67.5	11.4	13.82
37	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Pi	Neutral	39	2/33/4	1507.0	70.3	5.1	1.52
38	Gemma4-26B-A4B	SGLang/BF16/on	Pi	Neutral	41	6/28/7	1502.8	68.1	14.6	7.88
39	Granite-4.2-30B	SGLang/BF16/off	Pi	Neutral	39	1/34/4	1502.6	69.6	2.6	8.71
40	Qwen3.8-27B	SGLang/BF16/low	Codex	Detailed	39	2/34/3	1502.4	69.6	5.1	35.91
41	Nemotron3.5-Lightning	SGLang/BF16/off	Pi	Neutral	41	0/40/1	1501.2	68.3	0.0	0.67
42	Qwen3.8-27B	SGLang/FP8/xhigh	Codex	Neutral	39	0/36/3	1499.7	70.2	0.0	85.56
43	Qwen3.8-27B	SGLang/FP8/low	Pi	Neutral	40	5/27/8	1499.3	69.7	12.5	32.57
44	Qwen3.8-27B	vLLM/FP8/low	Pi	Detailed	42	5/29/8	1495.8	68.1	11.9	22.02
45	Qwen3.8-27B	SGLang/FP8/xhigh	Pi	Detailed	40	0/39/1	1493.6	68.6	0.0	52.46
46	Gemma4-26B-A4B	vLLM/BF16/off	Pi	Detailed	39	0/36/3	1489.9	70.4	0.0	5.93
47	Gemma4-26B-A4B	SGLang/BF16/off	Pi	Neutral	41	0/40/1	1489.2	68.4	0.0	1.36
48	Qwen3.8-27B	Transformers/FP8/low	Reference	Neutral	41	0/37/4	1487.9	68.6	0.0	107.55

Table 9: **Runtime service–harness–strategy prompt results, ranks 49–96.** Configuration gives engine / precision / reasoning. W/D/L denotes wins/draws/losses. All technical-loss counts are zero. Latency (s) follows Eq. (9).

Rank	Model	Configuration	Harness	Prompt	Games	W/D/L	Rating	RD	WR %	Latency (s)
49	Gemma4-26B-A4B	vLLM/BF16/off	Codex	Neutral	40	0/35/5	1487.6	70.1	0.0	10.43
50	Gemma4-26B-A4B	vLLM/BF16/off	Reference	Detailed	41	2/34/5	1486.0	69.2	4.9	1.03
51	Ornith-1.5-35B-A3B	SGLang/FP8/off	Pi	Neutral	44	4/37/3	1484.6	68.5	9.1	1.94
52	Gemma4-26B-A4B	SGLang/BF16/on	Pi	Detailed	46	6/36/4	1481.4	66.9	13.0	8.14
53	Qwen3.8-27B	vLLM/FP8/low	Pi	Neutral	41	2/36/3	1478.7	67.9	4.9	21.06
54	Qwen3.6-35B-A3B	SGLang/FP8/off	Codex	Detailed	40	4/34/2	1478.7	68.7	10.0	2.50
55	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Codex	Detailed	40	3/36/1	1473.9	68.5	7.5	3.93
56	Qwen3.8-27B	SGLang/BF16/low	Reference	Detailed	39	7/27/5	1473.2	68.7	17.9	26.94
57	Qwen3.8-27B	SGLang/BF16/low	Reference	Neutral	39	4/24/11	1469.4	70.5	10.3	26.34
58	Qwen3.8-27B	Transformers/FP8/low	Codex	Neutral	40	0/35/5	1464.8	69.0	0.0	137.95
59	Qwen3.8-27B	SGLang/BF16/low	Pi	Neutral	40	7/25/8	1461.4	68.9	17.5	23.53
60	Muse-Glimmer-30B	SGLang/BF16/low	Codex	Detailed	44	4/34/6	1459.2	67.3	9.1	20.24
61	Granite-4.2-30B	SGLang/BF16/off	Pi	Detailed	40	0/38/2	1457.7	69.4	0.0	6.99
62	Qwen3.8-27B	SGLang/FP8/low	Reference	Detailed	39	2/33/4	1456.8	69.1	5.1	37.90
63	Qwen3.8-27B	SGLang/FP8/off	Codex	Neutral	39	5/31/3	1452.5	69.2	12.8	10.73
64	Qwen3.8-27B	vLLM/FP8/low	Codex	Detailed	39	4/31/4	1448.8	69.3	10.3	35.07
65	Qwen3.8-27B	SGLang/FP8/low	Reference	Neutral	39	1/32/6	1448.6	69.9	2.6	35.00
66	Nemotron3.5-Lightning	SGLang/BF16/off	Codex	Neutral	40	1/34/5	1448.0	68.4	2.5	2.75
67	Nemotron3.5-Lightning	SGLang/BF16/off	Reference	Detailed	42	3/30/9	1445.3	67.7	7.1	1.61
68	Qwen3.8-27B	SGLang/FP8/xhigh	Codex	Detailed	39	0/37/2	1443.3	68.5	0.0	85.85
69	Gemma4-26B-A4B	SGLang/BF16/off	Pi	Detailed	40	0/39/1	1442.8	68.6	0.0	1.13
70	Qwen3.8-27B	SGLang/FP8/xhigh	Reference	Neutral	39	0/39/0	1440.9	69.2	0.0	62.42
71	Qwen3.8-27B	SGLang/FP8/xhigh	Reference	Detailed	39	0/36/3	1438.9	68.7	0.0	63.07
72	Qwen3.8-27B	Transformers/FP8/low	Reference	Detailed	40	0/35/5	1438.9	68.7	0.0	115.54
73	Qwen3.8-27B + DSpark	SGLang/FP8/low	Codex	Detailed	40	6/25/9	1438.5	69.0	15.0	21.53
74	Nemotron3.5-Lightning	SGLang/BF16/off	Pi	Detailed	39	0/37/2	1436.1	68.5	0.0	1.84
75	Qwen3.8-27B	SGLang/FP8/off	Codex	Detailed	40	0/39/1	1433.3	68.3	0.0	10.95
76	Qwen3.8-27B	vLLM/FP8/low	Reference	Neutral	39	5/27/7	1432.6	69.5	12.8	26.23
77	Qwen3.8-27B	SGLang/BF16/low	Codex	Neutral	39	3/31/5	1420.7	69.5	7.7	31.82
78	Qwen3.8-27B	Transformers/FP8/low	Pi	Detailed	39	0/34/5	1418.0	68.7	0.0	108.61
79	Gemma4-26B-A4B	vLLM/BF16/off	Codex	Detailed	39	0/36/3	1416.9	69.7	0.0	16.38
80	Granite-4.2-30B	SGLang/BF16/off	Codex	Neutral	42	0/37/5	1416.6	67.3	0.0	13.35
81	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Reference	Neutral	39	1/35/3	1416.4	68.7	2.6	8.85
82	Ornith-1.5-35B-A3B	SGLang/FP8/off	Pi	Detailed	39	0/35/4	1414.6	70.8	0.0	2.20
83	Qwen3.8-27B	vLLM/FP8/low	Codex	Neutral	42	3/29/10	1413.7	68.0	7.1	32.15
84	Qwen3.8-27B	SGLang/FP8/low	Codex	Detailed	40	0/38/2	1412.8	68.3	0.0	53.37
85	Granite-4.2-30B	SGLang/BF16/off	Reference	Neutral	39	2/33/4	1411.3	69.1	5.1	9.80
86	Gemma4-26B-A4B	vLLM/BF16/off	Pi	Neutral	40	0/37/3	1411.0	69.2	0.0	23.03
87	Granite-4.2-30B	SGLang/BF16/off	Reference	Detailed	39	0/35/4	1406.9	70.2	0.0	10.41
88	Nemotron3.5-Lightning	SGLang/BF16/off	Codex	Detailed	40	4/30/6	1405.4	68.2	10.0	2.75
89	Qwen3.8-27B	SGLang/FP8/low	Codex	Neutral	39	2/31/6	1400.1	69.4	5.1	41.93
90	Qwen3.8-27B	Transformers/FP8/low	Codex	Detailed	40	0/35/5	1391.8	68.3	0.0	153.02
91	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Pi	Detailed	39	1/33/5	1391.1	68.9	2.6	1.68
92	Qwen3.8-27B	Transformers/FP8/low	Pi	Neutral	39	0/33/6	1380.3	69.5	0.0	93.77
93	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	Reference	Detailed	40	4/27/9	1377.3	68.9	10.0	8.26
94	Muse-Glimmer-30B	SGLang/BF16/low	Reference	Detailed	39	0/28/11	1340.0	71.6	0.0	13.39
95	Nemotron3.5-Lightning	SGLang/BF16/off	Reference	Neutral	39	1/27/11	1338.3	70.8	2.6	1.48
96	Muse-Glimmer-30B	SGLang/BF16/low	Reference	Neutral	39	0/29/10	1336.8	70.3	0.0	12.91

Table 10: **Arithmetic decomposition of the 96 fitted Ratings.** Shares partition total squared deviation from the grand mean.

Component	df	Sum of squares	Share %
Runtime service	15	387237.9	45.85
Harness	2	21881.0	2.59
Strategy prompt	1	4540.0	0.54
Runtime service–harness	30	332048.5	39.32
Runtime service–strategy prompt	15	29624.2	3.51
Harness–strategy prompt	2	6092.8	0.72
Three-way interaction	30	63141.8	7.48

E.3 Subscription-Based Services

Tables 11 and 12 report performance and access for all 24 configurations. Eleven have full measured availability. Among the others, the availability adjustment can be substantial. For example, Bailian Qwen3.7 Max moves from first by Rating to ninth by ESR, and Bailian DeepSeek V4 Flash moves from second to fourth. Even within Bailian, measured availability ranges from 35.5% for GLM-5.2 to 89.7% for Qwen3.8 Flash, underscoring the need to evaluate each service configuration.

Access failures and in-game outcomes. Bailian Qwen3.8 Max starts 24 of forty valid opportunities and records sixteen quota rejections. Its accepted games contain sixteen draws and eight losses, including two technical losses. Those losses enter Rating, whereas pre-start rejections enter availability. Its accepted-game logs contain 5.37 million input and 1.40 million output tokens. This usage excludes other account activity and does not by itself determine a plan's quota threshold.

Outcome profiles across experiments. The runtime, factorial, and subscription ladders have draw fractions of 44.7%, 75.5%, and 51.7%, respectively. Their entrant pools and realized opponents differ, so these rates describe each experiment rather than a controlled comparison of harness quality. Both local ladders contain zero participant-attributed technical losses, whereas the subscription sample contains 39.

Table 11: **Subscription-service results, ordered by ESR.** W/D/L denotes wins/draws/losses. TL denotes technical losses included in L. Availability is the percentage of resolved valid opportunities accepted.

Rank	Provider	Model	Games	W/D/L	TL	Rating	RD	WR %	Avail.%	ESR
1	Volcengine	Doubao Seed 2.0 Lite	260	122/92/46	0	1675.1	62.5	46.9	100.0	1675.1
2	Volcengine	DeepSeek V4 Flash	267	128/83/56	0	1641.4	62.3	47.9	100.0	1641.4
3	Volcengine	Ark Code Latest	243	82/106/55	0	1603.7	62.9	33.7	100.0	1603.7
4	Bailian	DeepSeek V4 Flash	59	27/16/16	1	1697.8	66.2	45.8	79.7	1573.3
5	Volcengine	MiniMax M3	246	79/111/56	0	1560.5	62.3	32.1	100.0	1560.5
6	Bailian	Qwen3.8 Flash	104	44/29/31	1	1606.9	62.5	42.3	89.7	1557.5
7	MiMo	MiMo V2.5 Pro	80	28/29/23	1	1597.9	63.7	35.0	85.1	1529.5
8	Kimi	Kimi K3	49	21/8/20	8	1609.7	66.5	42.9	83.1	1529.4
9	Bailian	Qwen3.7 Max	34	18/5/11	1	1713.7	74.6	52.9	66.7	1511.0
10	Volcengine	Doubao Seed Evolving	239	70/105/64	0	1507.5	61.6	29.3	100.0	1507.5
11	MiniMax	MiniMax M2.5	210	46/119/45	0	1518.1	61.8	21.9	93.8	1495.3
12	Volcengine	Doubao Seed 2.1 Turbo	237	45/135/57	0	1460.7	61.5	19.0	100.0	1460.7
13	MiniMax	MiniMax M3	241	53/123/65	0	1443.4	61.6	22.0	100.0	1443.4
14	MiniMax	MiniMax M2.7	214	18/158/38	0	1444.0	61.8	8.4	99.5	1442.6
15	GLM	GLM-5.3 Flash	209	17/141/51	1	1424.6	62.2	8.1	100.0	1424.6
16	MiMo	MiMo V2.5	160	24/92/44	1	1430.1	61.3	15.0	95.2	1416.2
17	Volcengine	DeepSeek V4 Pro	215	16/145/54	0	1387.5	61.6	7.4	100.0	1387.5
18	Volcengine	Kimi K3	153	7/115/31	6	1373.5	61.6	4.6	100.0	1373.5
19	Volcengine	GLM-5.3	192	9/136/47	3	1346.4	62.8	4.7	100.0	1346.4
20	GLM	GLM-5.3	55	8/23/24	9	1371.1	66.6	14.5	88.7	1341.2
21	Bailian	DeepSeek V4 Pro	55	3/40/12	1	1309.1	65.7	5.5	79.7	1258.3
22	Kimi	Kimi K2.7 Code	43	4/27/12	4	1300.7	68.5	9.3	78.2	1246.3
23	Bailian	Qwen3.8 Max	24	0/16/8	2	1281.6	83.9	0.0	60.0	1174.1
24	Bailian	GLM-5.2	11	1/6/4	0	1238.2	125.0	9.1	35.5	1035.0

Table 12: **Subscription-service opportunity and usage accounting.** Ranks identify the configurations in Table 11. Failures list quota/rate-limit/unavailable counts. S is the rating-derived score and U is delivered utility. Tokens are in millions.

Rank	Offered	Started	Failures	S	U	Input tokens	Output tokens
1	260	260	0/0/0	0.733	0.733	345.555	13.763
2	267	267	0/0/0	0.693	0.693	118.313	18.192
3	243	243	0/0/0	0.645	0.645	60.173	10.945
4	74	59	15/0/0	0.757	0.604	29.714	4.150
5	246	246	0/0/0	0.586	0.586	72.949	33.459
6	116	104	12/0/0	0.649	0.582	85.877	9.807
7	94	80	14/0/0	0.637	0.542	31.698	3.396
8	59	49	10/0/0	0.653	0.542	12.556	1.413
9	51	34	17/0/0	0.774	0.516	12.077	2.130
10	239	239	0/0/0	0.511	0.511	61.242	10.939
11	224	210	2/12/0	0.526	0.493	309.552	16.040
12	237	237	0/0/0	0.444	0.444	43.774	15.699
13	241	241	0/0/0	0.419	0.419	59.396	27.895
14	215	214	0/0/1	0.420	0.418	136.052	11.555
15	209	209	0/0/0	0.393	0.393	48.474	16.267
16	168	160	6/0/2	0.401	0.382	101.209	10.337
17	215	215	0/0/0	0.344	0.344	21.653	16.421
18	153	153	0/0/0	0.326	0.326	27.715	9.167
19	192	192	0/0/0	0.292	0.292	48.249	32.641
20	62	55	7/0/0	0.323	0.286	20.179	6.864
21	69	55	14/0/0	0.250	0.199	5.309	4.188
22	55	43	12/0/0	0.241	0.188	5.328	1.344
23	40	24	16/0/0	0.221	0.133	5.371	1.398
24	31	11	20/0/0	0.181	0.064	4.011	1.581

F Protocol Comparison Details

F.1 Conditions and Interpretation

Each comparison uses Reference/Neutral and 600 games. Single actor and turn-based retain the sixteen Default services, whereas image observations use twelve vision-capable services. Default is the main runtime ladder, reused for comparison. Matchmaking is adaptive, and the comparison runs use dynamic GPU allocation instead of Default's resident services. The results therefore describe performance under each protocol rather than an isolated causal effect.

Table 13: **Outcome profiles across protocol settings.** Each setting contains 600 games and no participant-attributed technical losses. Default reuses the main runtime ladder.

Condition	Draws/games	Draw %
Default	268/600	44.7
Single actor	270/600	45.0
Image	512/600	85.3
Turn-based	362/600	60.3

F.2 Multi-Agent versus Single Actor

Single actor limits each side to one session and disables delegation and communication. In Default, child sessions share their side's runtime capacity. Qwen3.6 and Qwen3.8/DSpark never use children, whereas Ornith does so in 108 of 117 games. Gemma4/SGLang uses children in ten of 96 games with reasoning off and one of 116 with reasoning on. These differences show that enabling delegation does not imply that each model uses it.

Ornith rises from third in Default to second with one actor. Its win fraction increases from 47.0% to 59.8%, and its loss fraction falls from 47.0% to 34.6%. Mean sessions fall from 1.87 to 1.00 and latency from 2.53 to 1.62 seconds. Against Gemma4/SGLang off, Ornith's wins/draws/losses change from 11/4/14 to 13/2/5. Against Gemma4/SGLang on, they change from 20/2/17 to 20/2/8. Its improvement therefore includes fewer losses in these matchups. These results show that allowing multiple actors changes Ornith's organizational behavior and is accompanied by a substantial difference in performance. Child adoption in Table 14 is the fraction of games with an active child. "Sessions" denotes the average of each game's time-averaged concurrency, and peak is the largest observed session count.

F.3 Text versus Image Observations

Each image entrant plays 100 games. Gemma4/SGLang with reasoning on records 33 wins, 63 draws, and four losses. Qwen3.6 records five wins, 91 draws, and four losses. Across the image ladder, 512 of 600 games are draws and no game ends in a participant-attributed technical loss. The image ladder logs 59,535 rejected decisions among 222,015 completed requests. This measures action validity rather than visual comprehension.

The image ladder contains twelve vision-capable services, compared with sixteen in Default, so the ranking comparison also reflects differences in the participant pool.

A recorded visual-state interpretation error. In a recorded match against Qwen3.8/vLLM with low reasoning, Qwen3.6 controls the red side. At observation tick 18, it misidentifies the Capital at row 2, column 5 as an enemy target. The supplied image instead identifies that cell as its own Capital. The move from $(3, 5)$ to $(2, 5)$ transfers friendly troops. The match ends in a draw. This example highlights the difficulty of interpreting ownership directly from images.

Table 14: **Organization and execution behavior in Default.** All rows use the Reference harness and Neutral prompt. Sessions is the average of per-game mean concurrency, and Peak is the maximum concurrent session count. Child games have active child sessions. Decisions include non-movement actions. Rejected % counts recorded validation failures per completed request.

Model	Configuration	Games	Child games	Child %	Sessions	Peak	Dec./1,000 ticks	Rejected %
Qwen3.6-35B-A3B	SGLang/FP8/off	113	0	0.0	1.00	1	268.7	3.5
Gemma4-26B-A4B	SGLang/BF16/off	96	10	10.4	1.07	2	242.6	0.0
Ornith-1.5-35B-A3B	SGLang/FP8/off	117	108	92.3	1.87	2	386.1	7.4
Gemma4-26B-A4B	SGLang/BF16/on	116	1	0.9	1.00	2	85.4	0.0
Qwen3.8-27B + DSpark	SGLang/FP8/low	77	0	0.0	1.00	1	55.4	0.0
Qwen3.8-27B	SGLang/FP8/off	63	3	4.8	1.10	9	106.1	0.0
Qwen3.8-27B	vLLM/FP8/low	66	0	0.0	1.00	1	31.0	0.0
Qwen3.8-27B	SGLang/BF16/low	69	0	0.0	1.00	1	32.6	0.0
Qwen3.8-27B	SGLang/FP8/low	60	0	0.0	1.00	1	24.1	0.0
Muse-Glimmer-30B	SGLang/BF16/low	57	21	36.8	1.27	2	29.8	63.4
Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	61	8	13.1	1.12	2	40.3	7.0
Gemma4-26B-A4B	vLLM/BF16/off	65	0	0.0	1.00	1	186.1	30.9
Granite-4.2-30B	SGLang/BF16/off	57	2	3.5	1.04	4	77.9	0.1
Qwen3.8-27B	SGLang/FP8/xhigh	60	10	16.7	1.09	2	13.3	0.2
Qwen3.8-27B	Transformers/FP8/low	59	0	0.0	1.00	1	8.7	0.0
Nemotron3.5-Lightning	SGLang/BF16/off	64	2	3.1	1.03	2	302.5	15.8

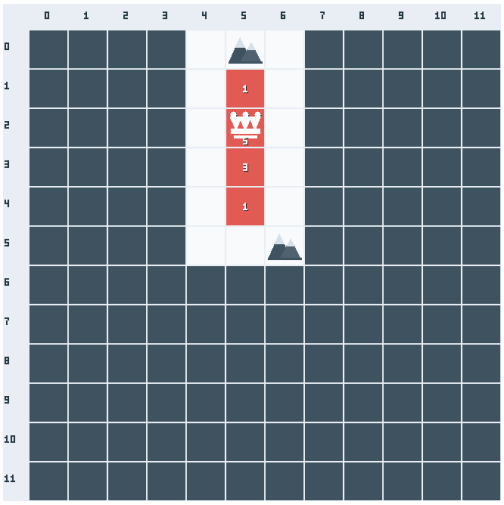


Figure 8: **A visual-state interpretation error.** Observation supplied to Qwen3.6 at tick 18. The red Capital at row 2, column 5 belongs to the acting player, although Qwen3.6 misidentifies it as an enemy target. Coordinates are zero-based.

F.4 Real-Time versus Turn-Based Execution

Turn-based execution requests one action from each living actor, then submits the responses together after all requests finish and the minimum ten-tick stride is met. Game time and troop growth continue during this wait. Compared with Default, this changes how response speed translates into action opportunities. Qwen3.6 and Qwen3.8/DSpark process 44.6 and 46.2 decisions per 1,000 ticks, respectively, compared with 268.7 and 55.4 in Default.

Reasoning-enabled configurations occupy the top five turn-based positions. Qwen3.8/DSpark leads, followed by Qwen3.8/vLLM low and Gemma4/SGLang on. Within Qwen3.8/SGLang/FP8, low reasoning yields 15 wins, 52 draws, and two losses, while xhigh yields two wins, 69 draws, and no losses. Their latency summaries are 34.04 and 58.73 seconds. Because both sides wait for all requests to finish, xhigh reasoning delays joint action submission and lengthens the time needed to resolve a match, making timeout draws more likely.

F.5 Complete Results

Tables 15 to 17 report all three comparison conditions. Default results are in Table 7, and organization statistics are in Table 14.

Table 15: **Complete single-actor results.** All rows use the Reference harness and Neutral prompt. Configuration gives engine / precision / reasoning. Child % is the fraction of games with an active child. Sessions is mean concurrency. Latency (s) follows Eq. (9).

Rank	Model	Configuration	Games	W/D/L	Rating	RD	Child %	Sessions	Latency (s)
1	Qwen3.6-35B-A3B	SGLang/FP8/off	107	59/11/37	1838.8	62.5	0.0	1.00	1.51
2	Ornith-1.5-35B-A3B	SGLang/FP8/off	107	64/6/37	1824.1	61.7	0.0	1.00	1.62
3	Gemma4-26B-A4B	SGLang/BF16/off	93	39/24/30	1784.5	62.1	0.0	1.00	2.55
4	Gemma4-26B-A4B	SGLang/BF16/on	100	41/12/47	1683.5	62.1	0.0	1.00	9.26
5	Qwen3.8-27B + DSpark	SGLang/FP8/low	82	38/18/26	1611.2	64.8	0.0	1.00	12.95
6	Qwen3.8-27B	SGLang/BF16/low	65	16/35/14	1486.8	64.5	0.0	1.00	24.51
7	Qwen3.8-27B	SGLang/FP8/off	69	14/38/17	1480.0	64.4	0.0	1.00	7.56
8	Qwen3.8-27B	vLLM/FP8/low	72	19/31/22	1471.8	65.1	0.0	1.00	23.27
9	Gemma4-26B-A4B	vLLM/BF16/off	67	8/44/15	1422.6	63.5	0.0	1.00	1.26
10	Granite-4.2-30B	SGLang/BF16/off	59	2/53/4	1409.2	64.1	0.0	1.00	10.28
11	Qwen3.8-27B	SGLang/FP8/low	64	7/47/10	1393.0	64.4	0.0	1.00	32.91
12	Qwen3.8-27B	SGLang/FP8/xhigh	61	4/49/8	1368.3	64.3	0.0	1.00	53.43
13	Muse-Glimmer-30B	SGLang/BF16/low	64	6/44/14	1367.4	63.6	0.0	1.00	13.89
14	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	63	8/46/9	1345.7	63.8	0.0	1.00	8.66
15	Qwen3.8-27B	Transformers/FP8/low	60	1/47/12	1266.4	64.5	0.0	1.00	103.87
16	Nemotron3.5-Lightning	SGLang/BF16/off	67	4/35/28	1232.6	67.2	0.0	1.00	1.60

Table 16: **Complete image-observation results.** All rows use the Reference harness and Neutral prompt. Configuration gives engine / precision / reasoning. Child % is the fraction of games with an active child. Sessions is mean concurrency. Latency (s) follows Eq. (9).

Rank	Model	Configuration	Games	W/D/L	Rating	RD	Child %	Sessions	Latency (s)
1	Gemma4-26B-A4B	SGLang/BF16/on	100	33/63/4	1600.4	61.9	12.0	1.10	14.12
2	Qwen3.8-27B + DSpark	SGLang/FP8/low	100	13/80/7	1532.5	60.8	0.0	1.00	18.99
3	Gemma4-26B-A4B	SGLang/BF16/off	100	10/84/6	1516.4	60.6	1.0	1.01	1.23
4	Qwen3.8-27B	vLLM/FP8/low	100	13/82/5	1515.1	60.8	0.0	1.00	27.82
5	Ornith-1.5-35B-A3B	SGLang/FP8/off	100	7/80/13	1510.3	60.7	8.0	1.04	1.57
6	Qwen3.6-35B-A3B	SGLang/FP8/off	100	5/91/4	1504.8	60.6	20.0	1.20	1.59
7	Gemma4-26B-A4B	vLLM/BF16/off	100	2/92/6	1499.5	60.7	0.0	1.00	0.81
8	Qwen3.8-27B	Transformers/FP8/low	100	0/92/8	1493.9	60.8	0.0	1.00	102.78
9	Qwen3.8-27B	SGLang/BF16/low	100	2/89/9	1488.8	60.7	2.0	1.02	32.71
10	Qwen3.8-27B	SGLang/FP8/xhigh	100	0/92/8	1482.4	60.7	4.0	1.02	104.73
11	Qwen3.8-27B	SGLang/FP8/off	100	2/89/9	1479.4	60.7	4.0	1.01	5.18
12	Qwen3.8-27B	SGLang/FP8/low	100	1/90/9	1477.8	60.7	1.0	1.01	47.33

Table 17: **Complete turn-based results.** All rows use the Reference harness and Neutral prompt. Configuration gives engine / precision / reasoning. Child % is the fraction of games with an active child. Sessions is mean concurrency. Latency (s) follows Eq. (9).

Rank	Model	Configuration	Games	W/D/L	Rating	RD	Child %	Sessions	Latency (s)
1	Qwen3.8-27B + DSpark	SGLang/FP8/low	78	44/14/20	1812.2	66.6	0.0	1.00	12.87
2	Qwen3.8-27B	vLLM/FP8/low	73	22/36/15	1723.0	63.4	0.0	1.00	24.94
3	Gemma4-26B-A4B	SGLang/BF16/on	81	28/23/30	1683.2	62.6	0.0	1.00	8.19
4	Qwen3.8-27B	SGLang/BF16/low	71	14/47/10	1591.9	63.8	1.4	1.00	25.49
5	Qwen3.8-27B	SGLang/FP8/low	69	15/52/2	1588.1	63.7	0.0	1.00	34.04
6	Ornith-1.5-35B-A3B	SGLang/FP8/off	80	24/39/17	1534.9	63.8	43.8	1.37	2.04
7	Qwen3.6-35B-A3B	SGLang/FP8/off	94	30/37/27	1514.5	64.3	1.1	1.01	1.29
8	Gemma4-26B-A4B	SGLang/BF16/off	76	18/41/17	1460.6	62.7	1.3	1.01	2.18
9	Qwen3.8-27B	Transformers/FP8/low	66	0/66/0	1432.6	62.8	0.0	1.00	109.40
10	Qwen3.8-27B	SGLang/FP8/xhigh	71	2/69/0	1424.1	63.0	23.9	1.16	58.73
11	Laguna-XS-2.1-33B-A3B	SGLang/FP8/on	75	12/48/15	1413.8	62.7	12.0	1.08	7.95
12	Granite-4.2-30B	SGLang/BF16/off	72	5/54/13	1399.6	62.9	1.4	1.01	9.31
13	Muse-Glimmer-30B	SGLang/BF16/low	72	6/54/12	1380.8	62.4	19.4	1.13	13.33
14	Gemma4-26B-A4B	vLLM/BF16/off	72	4/53/15	1360.5	62.9	0.0	1.00	0.86
15	Qwen3.8-27B	SGLang/FP8/off	77	11/40/26	1352.4	63.2	2.6	1.02	7.56
16	Nemotron3.5-Lightning	SGLang/BF16/off	73	3/51/19	1330.2	63.0	1.4	1.01	1.11

G Qualitative Match Analyses

We examine four decisive games and four draws, comprising two runtime-service games, four runtime–harness–prompt games, and two subscription-service games. Cases are selected to cover different observable paths to an outcome, not to estimate their prevalence. Each has a linked public replay. We cross-check board transitions and processed actions against match records, distinguishing applied moves from inference requests that produce no applied action. Coordinates are zero-based. All four harness comparisons hold the runtime and prompt fixed within the game, but individual matches do not isolate a general harness effect.

G.1 Expansion followed by a concentrated attack

Match replay: `https://generalsarena.github.io/?view=records&evaluation=runtime-services&game=6`

In the runtime-service ladder, Qwen3.6/SGLang/FP8 with reasoning off defeats Qwen3.8/DSpark, both using Reference/Neutral. By tick 200, Qwen3.6 holds 44 cells against DSpark's sixteen. Qwen3.6 captures the City at $(6, 4)$ at tick 399, while DSpark captures another at tick 448. Expansion alone does not immediately end the game. Qwen3.6 also spends repeated moves transferring troops between two bottom-row cells before assembling its final attack.

The decisive sequence begins at tick 709. Qwen3.6 moves troops out of its Capital and combines them with the City garrison, then advances along row 5 to DSpark's Capital at $(4, 9)$, capturing it at tick 735. The final attacking cell holds 231 units immediately before the move, against forty at the target. DSpark's last applied move is at tick 700, moving a separate force northward. Across the game, Qwen3.6 executes 187 accepted moves and DSpark 42. The replay connects the outcome to a completed reinforcement-and-attack sequence after the opponent's last recorded move, rather than treating a larger action count as sufficient evidence of better play.

G.2 Frequent movement without decisive progress

Match replay: `https://generalsarena.github.io/?view=records&evaluation=runtime-services&game=1`

Qwen3.8/SGLang/FP8 with low reasoning draws against Gemma4/vLLM/BF16 with reasoning off under Reference/Neutral. Gemma executes 249 accepted moves against Qwen's 37, but many repeatedly redistribute small forces near the bottom-right corner. It makes 51 transfers from $(11, 10)$ to $(11, 11)$ and 62 from $(11, 11)$ to $(10, 11)$. Qwen instead captures the City at $(2, 4)$ at tick 399, yet its later movement also fails to reach the opposing Capital.

Both Capitals remain intact at tick 1679. Gemma retains 827 units at its Capital, while Qwen retains 329 at its Capital and 533 at the captured City. These reserves are not converted into a decisive attack. The request summary additionally records 155 decision failures for Gemma, separate from the processed actions shown in the replay. Thus interface failures coexist with substantial valid movement, and neither fact alone characterizes the game. This draw illustrates limited conversion of activity and accumulated resources into the stated objective, rather than a sustained contest over either Capital.

G.3 Surviving an early threat and completing a later attack

Match replay: `https://generalsarena.github.io/?view=records&evaluation=harness-strategy-interactions&game=35`

Two Gemma4/SGLang/BF16 configurations with reasoning off and the Detailed prompt face each other through Reference and Codex. Codex threatens the Reference Capital early. At tick 370, an attack leaves that Capital with only two units. The next attack at tick 380 carries too few troops to overcome its renewed garrison. Meanwhile, Reference captures a City at $(6, 4)$ at tick 392 and later develops a larger territorial and troop presence. The eventual winner therefore does not play a uniformly safe opening.

Reference's earlier attacks on the opposing Capital also fail to finish the game. Its successful attempt follows a larger mobilization from the City at tick 1016 and a sequence of transfers toward $(7, 9)$. It reaches the adjacent cell with 224 units and captures the Capital at tick 1087. Codex's final recorded moves instead alternate troops between $(11, 10)$ and $(11, 11)$. The contrast is visible in the sequence of objectives actually reached, despite shared model, runtime, and prompt. This case distinguishes threatening a Capital, sustaining sufficient force at that target, and eventually completing its capture.

G.4 A SMALLER TERRITORY SUPPORTS A SUCCESSFUL COUNTERATTACK

Match replay: `https://generalsarena.github.io/?view=records&evaluation=harness-strategy-interactions&game=148`

Qwen3.8/DSpark with Neutral plays through Pi against the same runtime and prompt through Reference. Pi captures the City at $(5, 4)$ at tick 591. Reference nevertheless expands farther, holding forty cells against Pi's eighteen at tick 1000. Its forces also reach cells adjacent to Pi's Capital. Pi retains the City and a substantial Capital garrison, so the territorial disparity does not imply that its immediate defenses have collapsed.

At tick 1562, Pi mobilizes the City's accumulated force and advances through $(5, 5)$, $(4, 5)$, $(3, 5)$, and $(3, 4)$ toward the Reference Capital at $(2, 3)$. Reference simultaneously advances near Pi's Capital, but moves from the adjacent $(6, 8)$ to $(7, 8)$ at tick 1613. Pi completes its attack at tick 1628. It wins with 64 accepted moves against Reference's 83 and with less territory immediately before the capture. The sequence illustrates why broad expansion and action volume can coexist with defeat when another force completes a direct Capital attack first.

G.5 DELEGATION AND MESSAGING WITHOUT A SECOND MOVEMENT STREAM

Match replay: `https://generalsarena.github.io/?view=records&evaluation=harness-strategy-interactions&game=13`

Ornith/SGLang/FP8 with reasoning off and Detailed draws in a Pi–Codex matchup. Pi creates a child at tick 1 and records nineteen team messages. However, the child's two processed actions are messages, and all 44 accepted moves on that side come from the root. Codex uses only its root and makes 81 accepted moves. This is a concrete difference in organization under a shared runtime and prompt, but the extra session does not itself establish parallel movement or useful division of labor.

Neither side captures a City or Capital. Codex's last applied move occurs at tick 349, while Pi continues occasional expansion through tick 1656. The request summaries record additional inference activity, so the absence of later applied moves is not evidence that inference stopped. At the end, the sides hold 37 and 36 cells, with both Capitals intact. This draw has a different observable character from a defended Capital assault. Session creation and communication are present, but they do not develop into a coordinated attack in the recorded actions.

G.6 AN ACCUMULATED ADVANTAGE FAILS TO PRODUCE A CAPTURE

Match replay: `https://generalsarena.github.io/?view=records&evaluation=harness-strategy-interactions&game=241`

Qwen3.6/SGLang/FP8 with reasoning off and Neutral draws against itself through Reference and Codex. Reference captures two neighboring Cities at ticks 283 and 286, and a third at tick 967. At tick 983, it attacks the Codex Capital from $(7, 10)$ with 310 units in the source cell. The attack reduces the defending garrison from 458 to 149 without capturing it. This is a substantial offensive attempt, not a game in which both sides merely remain near their starting positions.

Eleven subsequent attacks from the same neighboring cell begin with only two or three units there. They fail to outpace the Capital's growth. Reference finishes with three Cities and 67 cells against Codex's thirty, yet the Capital survives and the game is drawn. The request summary also records 455 decision failures for Reference, which are not all visible as processed actions. The replay supports a narrower behavioral conclusion than overall incompetence or strong defensive play. A large resource advantage and access to the target coexist with fragmented follow-up attacks and unreliable conversion of requests into actions.

G.7 A DIRECT ATTACK SUCCEEDS WHILE THE OPPONENT DEVELOPS ELSEWHERE

Match replay: `https://generalsarena.github.io/?view=records&evaluation=subscription-services&game=28`

Bailian Qwen3.7 Max defeats Volcengine DeepSeek V4 Flash under Reference/Neutral. At tick 600, DeepSeek holds 25 cells against Qwen's eight. DeepSeek also captures a City at $(9, 8)$ at tick 649. Qwen has already launched a concentrated force from its Capital at tick 597 and moves it east along row 4. The two trajectories therefore pursue different immediate objectives while the game continues.

From ticks 634 to 682, Qwen's force takes a chain of lightly held cells leading to $(4, 9)$, then turns toward DeepSeek's Capital at $(6, 9)$. DeepSeek's intervening applied moves capture the City and expand elsewhere rather than intercepting this force. Qwen captures the Capital at tick 704 despite executing fewer accepted moves overall, 31 against 41. The larger territorial footprint and newly acquired City do not compensate for the unfinished defense of the decisive target. This accepted match illustrates competitive behavior after access has succeeded. It does not measure either service's ability to accept future match opportunities or explain its availability-adjusted ranking.

G.8 ACTIVE DELEGATION AND OPPOSING ADVANCES STILL END IN A DRAW

Match replay: `https://generalsarena.github.io/?view=records&evaluation=subscription-services&game=1464`

First-party MiniMax M3 draws against first-party Kimi K3 under Reference/Neutral. Kimi creates a child at tick 77, assigns two cells to it at tick 114, and later reclaims them. Unlike the Ornith case, this child contributes 38 accepted moves. Kimi captures Cities at ticks 337 and 994, while MiniMax captures another at tick 1206 and takes Kimi's first City at tick 1438. Both sides therefore alter the contested position materially during the game.

Late in the match, MiniMax advances to $(8, 4)$ near Kimi's Capital at $(9, 3)$. Kimi's child advances north along column 8, reaching $(3, 8)$ at tick 1678, two Manhattan steps from MiniMax's Capital at $(2, 9)$. Kimi's root meanwhile redistributes troops near its own Capital. Neither advance completes a capture before the match ends at tick 1679. This draw contains child movement toward the opposing Capital, contested production, and advances toward both decisive targets. Together with the preceding draws, it shows why a common draw score can cover substantially different trajectories, which can be inspected through the replay rather than inferred from the score alone.

Request and processed-action accounting. Table 18 reconciles the failure counts cited in Appx. G.2 and G.6 with the corresponding request summaries and public replay actions. For these two Reference sessions, completed requests equal processed decisions plus logged decision failures. Processed decisions include accepted non-movement commands and decisions rejected during action processing, so their count is not the number of successful moves. These are distinct accounting stages, not interchangeable failure totals.

Table 18: **Request and action counts for two cases.** Both rows use Reference. The columns give completed inference requests, recorded processed decisions, logged decision failures, accepted moves, and other processed decisions.

Case and runtime	Requests	Processed	Failures	Moves	Other
G.2 Gemma4/vLLM	470	315	155	249	66
G.6 Qwen3.6/SGLang	652	197	455	196	1

For G.2, Other comprises 27 accepted passes and 39 invalid-action rejections. For G.6, it is one accepted note update. The 155 and 455 failures come from request summaries and are separate from these processed-action outcomes. They should not be interpreted as counts of rejected moves visible in the replay.

H RUNTIME AND API CONFIGURATION

This appendix records artifact identifiers, frozen configuration values, and retained toolchain versions for the three main experiments. Version entries describe configuration pins or toolchain records rather than independently verified build identities for every service process. A parameter marked "omitted" was not supplied by the client and does not imply a particular server-side default.

H.1 MODEL ARTIFACTS AND LOCAL SAMPLING

Table 19 identifies the model repositories and immutable artifact revisions. Tables 20 and 21 map all sixteen local runtimes to these artifacts and application-side sampling settings, which agree across the runtime-service and factorial experiments. These artifact revisions identify model repositories, not serving-engine or harness commits.

The local model families are Qwen3.6-35B-A3B (Qwen Team, 2026a), Qwen3.8-27B (Qwen Team, 2026d), and Gemma4-26B-A4B (Gemma Team, 2026), together with Ornith-1.5 (Ornith Team, 2026), Muse Glimmer (Meta Superintelligence Labs, 2026), and Laguna-XS-2.1 (Poolside, 2026). Granite-4.2 (IBM Granite Team, 2026) and Nemotron3.5-Lightning (NVIDIA, 2026) complete the model set. The DSpark configuration additionally uses the Qwen3.8-27B-DSpark draft artifact (RadixArk, 2026). These model cards identify the model families and draft, while Table 19 specifies the exact repositories and revisions used in our experiments.

Table 19: **Pinned local model artifacts.** R10 is the separate BF16 DSpark draft. Repository identifiers and revisions specify the loaded artifacts.

ID	Repository	Revision
R01	`Qwen/Qwen3.8-27B-FP8`	`017b9c7af6b5689d5dd426a76e0bc077eb5ca20a`
R02	`Qwen/Qwen3.8-27B`	`1d4bf0f2ff6012fd82039f2fa52739d0dd7c60c0`
R03	`Qwen/Qwen3.6-35B-A3B-FP8`	`95a723d08a9490559dae23d0cff1d9466213d989`
R04	`unsloth/Muse-Glimmer-30B`	`55a49b91a33d176bc99db7569f678d4c64cd91a1`
R05	`google/gemma-4-26B-A4B-it`	`4d7ae4984b7db7de8f8457170b3f1a419ee76d52`
R06	`poolside/Laguna-XS-2.1-FP8`	`3fd8184a5b76f8788b525c1ac6c820f7314c456e`
R07	`ornith-ai/Ornith-1.5-35B-A3B-FP8`	`fab11c26e2325a42f4b32da0249c819a0bade1b1`
R08	`ibm-granite/granite-4.2-30b`	`70fc514076785017d93087f3d8b0676426b6b355`
R09	`nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16`	`a9904d24bcc1d289a1950fa9d2b978c47cf903b9`
R10	`RadixArk/Qwen3.8-27B-DSpark`	`85ef153be924f17ce4bf62726954eeaa4a73e854`

Table 20: **The sixteen local runtime configurations.** SGL, vLLM, and Trans denote the separately pinned serving environments in Table 22. The profile column refers to Table 21.

Runtime	Artifact	Engine	Precision	Reasoning	Profile
Qwen3.6-35B-A3B	R03	SGL	FP8	off	Q-off
Gemma4-26B-A4B	R05	SGL	BF16	off	Gemma
Ornith-1.5-35B-A3B	R07	SGL	FP8	off	Ornith
Gemma4-26B-A4B	R05	SGL	BF16	on	Gemma
Qwen3.8-27B + DSpark	R01 + R10	SGL	FP8 + BF16 draft	low	Q-reason
Qwen3.8-27B	R01	SGL	FP8	off	Q-off
Qwen3.8-27B	R01	vLLM	FP8	low	Q-reason
Qwen3.8-27B	R02	SGL	BF16	low	Q-reason
Qwen3.8-27B	R01	SGL	FP8	low	Q-reason
Muse-Glimmer-30B	R04	SGL	BF16	low	Muse
Laguna-XS-2.1-33B-A3B	R06	SGL	FP8	on	Laguna
Gemma4-26B-A4B	R05	vLLM	BF16	off	Gemma
Granite-4.2-30B	R08	SGL	BF16	off	G/N
Qwen3.8-27B	R01	SGL	FP8	xhigh	Q-reason
Qwen3.8-27B	R01	Trans	FP8 artifact	low	Q-reason
Nemotron3.5-Lightning	R09	SGL	BF16	off	G/N

Table 21: **Explicit local sampling profiles.** All profiles set min-p to 0 and repetition penalty to 1. Omitted values are not inferred from current model cards or current software defaults.

Profile	Temperature	Top-p	Top-k	Presence penalty
Q-off	0.7	0.8	20	1.5
Q-reason	1.0	0.95	20	0.0
Gemma	1.0	0.95	64	0.0
Ornith	0.6	0.95	20	0.0
Laguna	1.0	1.0	20	0.0
G/N	1.0	0.95	omitted	0.0
Muse	omitted	omitted	omitted	0.0

Qwen off/on comparisons also change the sampling profile, as shown in Table 21, and therefore do not isolate reasoning alone. Low and xhigh share the Q-reason profile.

Local clients use a configured context length of 65,536 tokens, an output allowance of 32,768 tokens, a request timeout of 1,800 seconds, and at most three adapter attempts. The output allowance and its harness-specific adjustment are described in Appx. D. Reasoning controls are passed through model-specific chat-template options. Qwen uses `enable_thinking` and, when enabled, `reasoning_effort`. Gemma, Ornith, Laguna, Granite, and Nemotron use `enable_thinking`, while Muse uses `reasoning_strength="low"`. Action calls use native tools with required tool choice and parallel tool calls disabled. No custom stop sequence or stop-token override is supplied. Local decision requests receive deterministic seeds, while hosted requests do not. The effective omitted sampling values for Muse and omitted top-k values for Granite and Nemotron were not recorded.

H.2 Serving Environments and Harness Versions

Table 22: **Separate local serving environments.** The Transformers dependency installed with each serving engine differs from the standalone Transformers server version.

Environment	Pinned version	Transformers dependency
SGLang	0.5.18	5.12.1
vLLM	0.28.0+cu129	5.15.1
Transformers Serve	5.16.1	5.16.1

All local runtimes use tensor parallelism one. Qwen3.8/SGLang specifies an FP8 E4M3 KV cache, FlashInfer attention, and BF16 Mamba state. Its ordinary and DSpark configurations set `mamba-full-memory-ratio` to 2.34 and 6.08, respectively. Muse uses its native `muse` reasoning and tool-call parsers in language-model-only mode. Gemma/vLLM uses the repository's `tool_chat_template_gemma4.v0.28.0.jinja`. Laguna uses `poolside_v1` parsers and the shared-expert tensor-parallel-one mode. Nemotron retains BF16 weights but explicitly uses float16 Mamba state. The standalone Transformers server uses automatic dtype selection with compilation and continuous batching disabled.

DSpark combines the FP8 main artifact R01 with the BF16 draft R10. Its launch enables `DSPARK` and FlashInfer draft attention. Retained server arguments record one speculative step, eight draft tokens, Eagle top-k one, and no additional draft quantization. These are parsed server settings rather than a claim that every value was explicitly passed on the command line.

The Pi dependency lock pins `pi-agent-core`, `pi-ai`, and `pi-coding-agent` from `@earendil-works` to 0.84.2. The retained Codex toolchain reports `codex-cli` 0.144.5. Both TypeScript bridge packages are version 0.1.0. Reference and the Python adapters are repository implementations without separate package releases. The retained locks and toolchain records do not provide a complete per-session binary identity audit. The runtime settings above therefore specify the recorded configuration without claiming complete process-level version coverage.

H.3 Subscription Endpoints and Request Settings

Table 23 lists the exact API aliases of all 24 subscription configurations. Displayed model names identify the evaluated service catalog. API aliases, particularly rolling aliases, do not establish immutable model weights or server software versions.

The subscription models include Qwen3.7 Max (Qwen Team, 2026b), Qwen3.8 Max (Qwen Team, 2026c), and Qwen3.8 Flash (Alibaba Cloud, 2026). DeepSeek V4 Pro and Flash are described in the shared V4 report (DeepSeek-AI, 2026). MiniMax M2.5 (MiniMax, 2026a), M2.7 (MiniMax, 2026b), and M3 (MiniMax, 2026c) have separate release articles, while Kimi K2.7 Code (Moonshot AI, 2026) and Kimi K3 (Kimi Team et al., 2026) have a dedicated model page and technical report, respectively. MiMo V2.5 and V2.5 Pro share a series release article (Xiaomi MiMo Team, 2026), and GLM-5.2 (Z.ai, 2026a), GLM-5.3 (Z.ai, 2026c), and GLM-5.3-Flash (Z.ai, 2026b) have separate release articles. Doubao Seed 2.0 Lite (ByteDance Seed Team, 2026a) and Seed 2.1 Turbo (ByteDance Seed Team, 2026b) are covered by their series releases. Seed Evolving is documented in the provider's model announcements (Volcengine, 2026), while Ark Code Latest is a routing alias documented in its service guide (Volcengine Service Assistant, 2026).

Table 23: **Subscription API aliases and routes.** All configurations use Reference/Neutral and OpenAI-compatible Chat Completions. Route suffixes identify the API families without disclosing account-specific addresses.

Provider / plan	Exact model aliases	Route suffix
Volcengine / Agent Plan	`doubao-seed-2.0-lite`, `ark-code-latest`, `minimax-m3`, `doubao-seed-2.1-turbo`, `deepseek-v4-flash`, `glm-5.3`, `doubao-seed-evolving`, `deepseek-v4-pro`, `kimi-k3`	`/api/plan/v3/chat/completions`
GLM / Coding Plan	`glm-5.3`, `glm-5.3-flash`	`/api/coding/paas/v4/chat/completions`
Kimi / Coding Plan	`k3`, `kimi-for-coding`	`/coding/v1/chat/completions`
MiniMax / Token Plan	`MiniMax-M3`, `MiniMax-M2.7`, `MiniMax-M2.5`	`/v1/chat/completions`
MiMo / Token Plan	`mimo-v2.5-pro`, `mimo-v2.5`	`/v1/chat/completions`
Bailian / Token Plan	`qwen3.8-max`, `qwen3.8-flash`, `qwen3.7-max`, `deepseek-v4-pro-0813`, `deepseek-v4-flash-0731`, `glm-5.2`	`/compatible-mode/v1/chat/completions`

Subscription requests specify `max_tokens=32768`, non-streaming responses, native action tools, `tool_choice="auto"`, and `parallel_tool_calls=false`, with a 330-second timeout. Temperature, top-p, top-k, reasoning-effort overrides, thinking switches, custom stop sequences, and decision seeds are not sent. Thus provider-default inference means the absence of these client-side overrides, not a shared or independently verified numerical default. Valid decisions must still satisfy tool-response and action parsing requirements.

Configured client context lengths are 204,800 for MiniMax M2.5/M2.7, and 1,000,000 for MiniMax M3, GLM 5.3/5.3 Flash, Bailian Qwen3.8 Max/Flash and GLM 5.2, and Volcengine MiniMax M3/GLM 5.3. Other configurations use 262,144. These are harness budgeting values rather than verified server-side hard limits. Recorded transport settings include maximum-attempt values of one and 64. These are distinct from the pre-match probe budget of at most three total attempts described in Appx. C.4. The provider's effective sampling values, immutable weight revisions, and server build identities are not recorded.

I LIMITATIONS AND FUTURE WORK

Available local GPU capacity bounds the runtime roster and configuration grid examined here. Building on dynamic runtime allocation, we plan to expand model loading and service scheduling to admit more local models and explore a wider range of inference configurations. The current three harnesses and two strategy prompts provide an initial view of their interactions. Adapting additional harnesses and developing more effective prompts would enable a broader analysis of how system design shapes competitive performance.

Our service study examines agent, coding, and token subscription plans, capturing performance under their access constraints. General-purpose commercial APIs, including pay-as-you-go services and alternative deployment arrangements, offer further settings to explore. Extending the evaluation to these access models would support comparisons across a wider range of service conditions, costs, and throughput levels.

The two-player arena provides a controlled setting for real-time competition on compact maps. Multi-player free-for-all matches, larger maps, and richer terrain would introduce additional demands on exploration, long-term planning, and multi-agent coordination. Exploring these settings would help identify which system behaviors persist across different forms of competition.