LARGE LANGUAGE MODEL-BASED GAME AGENTS ADJUST THEIR PLAY TO GRADED INSTRUCTIONS FOR COMPETING PRIORITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Automated playtesting may require agents to emulate human players at different skill levels, whereas game agents used as non-player characters may need to exhibit a consistent persona. These uses may call for different balances between playing strength and other desired behaviors. For example, in chess and shogi, preserving a win and favoring piece captures need not lead to the same move. We therefore investigate whether and how large language model (LLM)-based game agents respond to graded natural-language instructions specifying the relative importance of these goals. We test game agents built on three LLMs in Animal Shogi, Breakthrough, and Rush Hour on states where following the requested style costs playing strength. We vary relative priority over five levels using numerical, modal, and adverbial instructions. Across all tested models and games, agents exhibit a graded behavioral response, with style-matching actions becoming more frequent overall as the requested style is assigned higher priority.

1 INTRODUCTION

Automated play is now central to game development. Game agents automate play by choosing actions on behalf of players. For automated playtesting and quality assurance, developers may want agents that play at a desired level of strength while following a specified play style (Le Pelletier de Woillemont et al., 2022; Chen et al., 2025). Similar demands can arise for non-player characters (Ingram et al., 2023). For example, a developer may want an agent that plays like a beginner but favors attacks, or one that plays to win while taking bold risks. In each case, the agent must balance playing strength with the requested style. These two aims can favor different actions: in chess or shogi, a piece capture need not preserve a win. A useful game agent should therefore respond to the relative importance a user places on playing strength and the requested style.

Large language models (LLMs) have been used for both text generation and decision making in interactive environments, including games (Gallotta et al., 2024). Existing studies examine action selection and strategic reasoning in board and card games and game-theoretic tasks (Duan et al., 2024a;b), as well as the ability to pursue goals over many steps in interactive environments (Paglieri et al., 2025). These studies show that game performance depends on both the environment and the reasoning procedures used by the agent (Duan et al., 2024a;b). Together, they assess LLMs' strategic reasoning and goal-directed capabilities: how effectively agents choose actions, compete against opponents, and complete tasks. LLM agents may not only perform tasks well but also adjust their behavior in response to natural-language instructions (Ouyang et al., 2022; Wang et al., 2022).

This capability suggests that users could guide game agents by specifying the relative importance of playing strength and a requested play style. However, it remains unclear how effectively this instruction-following capability carries over to game agents. Here, instructions concern game behavior rather than text generation. They specify playing strength, which requires game understanding and complex thinking, and a play style that the agent must translate into concrete action choices. Therefore, we ask the following research question:

Research question. How do LLM-based game agents respond to graded user priorities over playing strength and a requested play style?

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.

We investigate whether agents respond to changes in these stated priorities and, if so, how this response manifests in their action choices. We vary the relative priority of playing strength and the requested style across five levels using numerical, modal, and adverbial instructions.

We examine game agents built on three LLMs in Animal Shogi, Breakthrough, and Rush Hour. We evaluate them at positions where following the requested style conflicts with playing strongly, because a style-matching action that also preserves the best outcome cannot reveal which priority drove the choice. Using exact solvers to assess game outcomes and programmatic rules to label style-matching actions, we select such conflict positions in all three games. In the main experiment, we present each selected position under different priority instructions and measure how the frequency of style-matching actions across positions changes with priority. In a separate analysis on a subset of positions, we repeat action selection at each priority level to investigate how agents respond to priority instructions within the same position. We also evaluate complete Rush Hour episodes, measuring style adherence and puzzle completion over the course of play.

Our contributions are:

- We evaluate game agents based on three LLMs across three games, five priority levels, and three natural-language formulations.
- We find a graded behavioral response: style-matching actions become more frequent as the requested style receives higher priority.
- To understand the overall pattern, we analyze repeated decisions at fixed positions and show that shifts in style-matching frequency within individual positions contribute to it.
- The response also extends to sequential play, where intermediate priorities often produce intermediate trade-offs between style adherence and task completion.

2 Related work

Game agents beyond strong play. Research on game agents also considers behaviors beyond playing strength. Human-compatible agents model human actions to serve as opponents or training partners (Devlin et al., 2016; Jacob et al., 2022). Other work learns recognizable styles for automated playtesting or varied player experiences (Le Pelletier de Woillemont et al., 2022; Ingram et al., 2023). More generally, multi-objective sequential decision making studies policies whose behavior depends on how several objectives are valued (Roijers et al., 2013). Conditional approaches train policies to respond to preference vectors or dynamic objective weights (Abels et al., 2019; Yang et al., 2019). These studies show that game behavior can be adapted to different styles and objective preferences, typically through game-specific training or other purpose-built mechanisms. Using LLMs may instead allow users to specify a desired play style and its importance relative to playing strength in natural language, without training a new policy for each request.

Instruction-based and graded control of LLM behavior. Instruction tuning established that language models can adapt to diverse requests expressed in natural language (Ouyang et al., 2022; Wang et al., 2022). Subsequent work studies whether prompts steer behavior toward specified personas or behavioral directions (Jiang et al., 2024; Li et al., 2024; Miehling et al., 2025). Beyond binary attributes, Zhou et al. study graded text generation and find that calibration depends on the intensity descriptions used (Zhou et al., 2024). Yamin et al. report that stated diagnostic costs often shift medical decisions in the intended direction, but not consistently across settings (Yamin et al., 2026). Multi-objective language policies can instead be trained to trade off conflicting objectives at inference time (Wang et al., 2024). These studies provide a basis for exploring whether game agents can act according to priorities expressed in natural language.

LLM-based game agents. Empirical studies of LLM-based game agents have largely focused on gameplay competence. ChessGPT combines language and policy learning for chess (Feng et al., 2023); ReTA and GTBench evaluate strategic reasoning in board, card, and game-theoretic settings (Duan et al., 2024a;b); and BALROG evaluates success in long-horizon interactive games (Paglieri et al., 2025). Related work on language-directed game behavior includes UniMaia, which adapts a frozen chess policy using a text encoder and policy-conditioning mechanism to control properties including openings and player strength (Siu & Istead, 2026). We test whether LLM game agents

change their action choices in response to the stated relative priority of a requested play style and playing strength.

3 LLM-BASED GAME AGENT WITH PRIORITY INSTRUCTIONS

Game agents may be expected to follow a specified play style while also playing strongly. These objectives need not favor the same action, and different uses may require different balances between them. We study whether LLM-based game agents change their action choices in response to graded natural-language instructions that vary the relative importance of the requested play style and the game objective. We first describe the setting, then introduce the priority instructions.

3.1 LLM GAME AGENT SETUP

We consider an LLM-based game agent. Given the game rules, current state, and available legal actions as in GameBench (Costarelli et al., 2024), together with an instruction specifying the relative priority of a requested play style and the game objective, the agent selects one action to execute. A requested play style specifies a property of the action the agent is asked to favor. The game objective is to win in the two-player games or solve the puzzle in as few actions as possible.

Each request consists of a system message followed by a user message. The system message contains the priority instruction. The user message presents the task rules and action notation, current state, numbered legal actions, selection request, and output format, in that order:

System message
`{priority instruction}`

User message
`{task rules and action notation}`
`{current state}`
`Legal actions: {numbered legal actions}`
`Select the best action.`
`thought: {reasoning}`
`action: {action identifier}`

Following GameBench, which reports improved game performance with chain-of-thought prompting, we use the `thought` field to explicitly request step-by-step reasoning to support deliberation before action selection (Costarelli et al., 2024). The selected action identifier follows in the `action` field. We extract this identifier from the response and use the selected action as the primary evaluation outcome.

3.2 PRIORITY INSTRUCTION FORMULATIONS

To vary the stated relative importance of the two objectives, we construct five ordered conditions. This use of graded instructions is motivated by prior work on graded behavioral control through natural-language instructions (Zhou et al., 2024; Yamin et al., 2026). The conditions change the two objectives' emphasis in opposite directions: as style emphasis decreases, game-objective emphasis increases. We instantiate this complementary ordering in three familiar ways of expressing graded importance: numerical percentages for explicit quantitative allocation, modal verbs for obligation or permission, and priority adverbs for qualitative preference strength. Table 1 summarizes the five paired conditions across these formulations.

Numerical priorities. We express the relative importance of the two objectives through complementary percentages:

`Goal priority: {S}% on {style}, {100-S}% on {game objective}.`

We use $S \in \{100, 75, 50, 25, 0\}$, spanning full style emphasis, equal emphasis, and full game-objective emphasis.

Modal priorities. We use modal verbs because they offer a familiar way to express degrees of requirement without numerical quantities. Style emphasis progresses from *must* through *should*, *can*, and *may* to a statement that no priority is assigned to the style, while the game-objective expression follows the reverse ordering. With M_g and M_s denoting the game-objective and style modals from Table 1, the template is:

`You {`M_g`} {game objective}.  You {`M_s`} {style}.`

Adverbial priorities. We use priority adverbs because they directly express degrees of preference without framing either objective as a requirement. Style emphasis progresses from *very strongly* through *strongly*, *moderately*, and *slightly* to no assigned priority, again paired with the reverse ordering for the game objective. In both verbal formulations, the game-objective sentence always precedes the style sentence. With A_g and A_s denoting the corresponding adverbs from Table 1, the template is:

`You {`A_g`} prefer {game objective}.  You {`A_s`} prefer {style}.`

In both verbal formulations, an objective assigned no priority instead receives an explicit no-priority sentence; Appendix A.3 gives the complete wording.

Table 1: Five priority levels expressed in three formulations. Each entry lists the style expression before the game-objective expression; numerical entries are percentages. The verbal entries summarize instructions that state the game objective before the style. Rows share an intended ordering, not an assumed semantic equivalence across formulations.

Style emphasis	Numerical (style/game obj.)	Modal (style/game obj.)	Adverbial (style/game obj.)
Highest	100/0	must / no priority	very strongly / no priority
High	75/25	should / may	strongly / slightly
Equal	50/50	can / can	moderately / moderately
Low	25/75	may / should	slightly / strongly
None	0/100	no priority / must	no priority / very strongly

4 Experimental Design

This section describes how we test whether changing the stated relative priority of a play style shifts an agent's action choices toward that style. We evaluate Claude Sonnet 4.5, Gemini 2.5 Flash, and GPT-5 mini in Animal Shogi, Breakthrough, and Rush Hour. We first define the requested styles for each game, then explain how we select positions and compare single decisions across priority conditions. Finally, we describe the complete-episode evaluation in Rush Hour.

4.1 Games and Styles

We use the two-player games Animal Shogi and Breakthrough alongside the single-player puzzle Rush Hour to examine priority responses across different game objectives and action spaces. In all three games, legal actions can be labeled for style matching, and exact solvers can assess the consequences of each action for the game objective. We briefly describe each game and its requested styles below. Detailed game rules appear in Appendix B.1.

Animal Shogi. Animal Shogi is a miniature form of shogi played by two players on a 3×4 board. We define three action-level styles. An action matches the *giraffe* style if it moves or drops the giraffe, the *backward* style if it moves a piece toward the acting player's home row, and the *capture* style if it captures an opposing piece.

Breakthrough. We use a two-player 5×5 variant of Breakthrough, a pawn race in which each player tries to reach the opponent's home row. An action matches the *capture* style if it captures an opposing pawn. We operationalize *right-flank* play as selecting a pawn currently in columns d–e,

irrespective of its destination. An action matches the *rear-advance* style if it advances one of the acting player's least advanced pawns.

Rush Hour. Rush Hour is a single-player sliding puzzle on a 6×6 grid in which vehicles must be moved to let the target car reach the exit. An action matches the *target-car* style if it slides the target car X, the *horizontal-vehicle* style if it slides any horizontal vehicle, and the *short-vehicle* style if it slides a vehicle of length two.

4.2 Single-Decision Evaluation

The single-decision evaluation tests whether stated priorities shift action choices between playing strongly and following a requested style. When a style-matching action also preserves the best game outcome, choosing it does not distinguish style-following from strong play. We therefore present the same conflict positions under different priority instructions. At a conflict position, at least one style-matching action is legal, yet choosing any such action would in fact forfeit a theoretical win in the two-player games or lengthen the shortest solution in Rush Hour. Figure 1 illustrates this condition with capture-style moves in Animal Shogi.

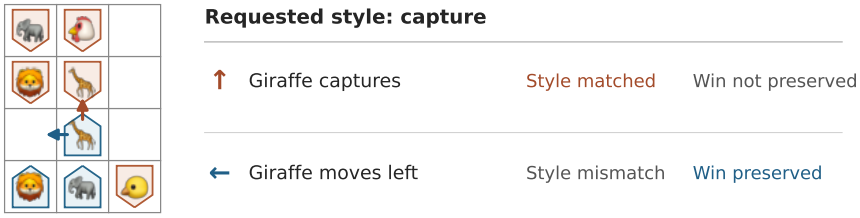

Figure 1: A style–strength conflict in Animal Shogi. Blue pieces belong to the agent and orange pieces to its opponent; animal emoji indicate piece types. Capturing matches the requested style but in fact forfeits a theoretical win, while moving left preserves it.

We identify these positions using exact solvers and action-level style labels. For each candidate position, a solver evaluates the state and every legal action, while a programmatic rule identifies style-matching actions. We retain positions satisfying the conflict criterion within three strata: forced wins in one, three, or five plies for Animal Shogi and Breakthrough, and exact solution distances of two, four, or six slides for Rush Hour. We continue until each style–stratum combination contains 100 Animal Shogi positions or 50 positions in Breakthrough and Rush Hour. The positions and labels are fixed before inference and shared across models and priority conditions. The exact-value computations are detailed in Appendix B.1.

At each selected position, we collect one response per model and condition: five levels in each of the numerical, modal, and adverbial formulations, plus a style-free baseline stating only the game objective. The models see neither solver values nor style labels. From the chosen action, we measure style matching (our primary outcome) and preservation of a theoretical win or shortest solution.

To examine whether the aggregate response reflects changes within individual positions, we repeat the numerical conditions at 25 Animal Shogi Giraffe positions shared across models. We obtain 100 responses per model, position, and priority level.

4.3 Sequential Play Evaluation

We test whether priority instructions change how often agents follow the requested style and solve Rush Hour puzzles over complete episodes. We use 15 held-out boards chosen separately from the conflict positions, and all models and conditions use the same boards. We test five numerical priority levels for each style, plus one style-free baseline shared across styles. Each model runs each board three times per condition.

At each turn, the model sees the current board and legal moves, but no move history or solver information. The instruction remains fixed until the puzzle is solved or the model reaches the larger

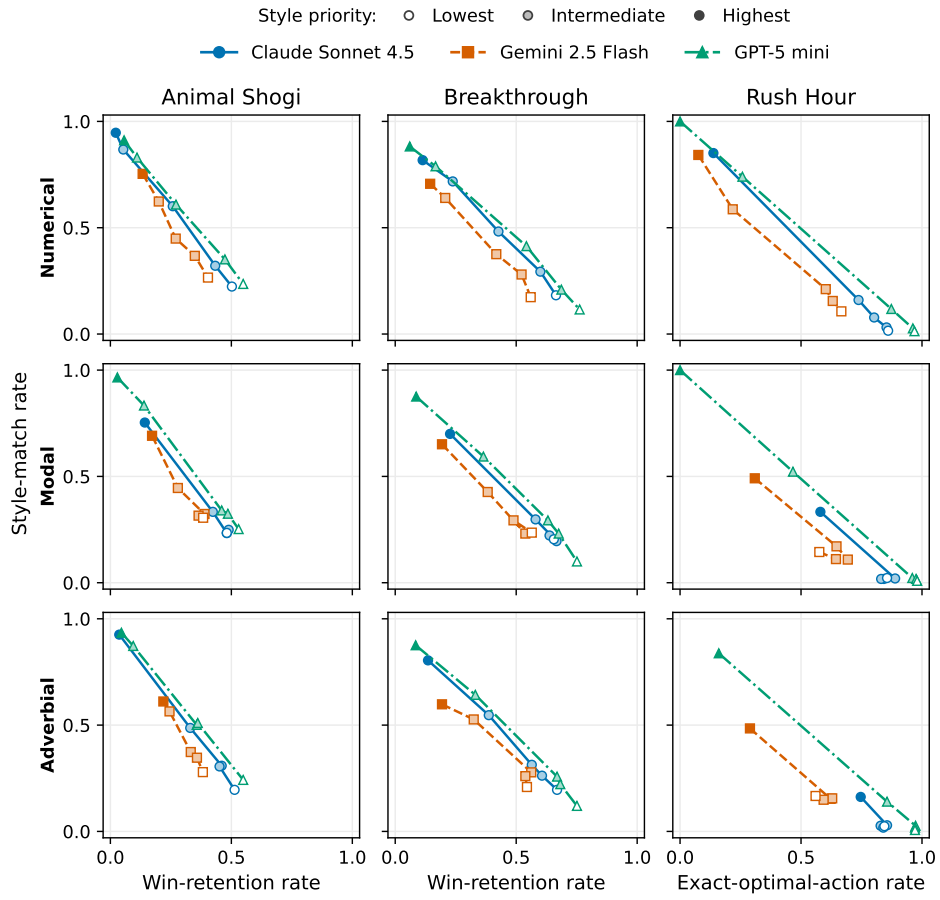

Figure 2: Observed style–strength trajectories across priority formulations, games, and models. Rows show numerical, modal, and adverbial instructions; columns show Animal Shogi, Breakthrough, and Rush Hour. Each line connects five priority conditions for one model, pooling three styles and three depth strata. The shared vertical axis shows style-match rate; the horizontal axis shows win retention in the two-player games and exact shortest-path optimality in Rush Hour. Marker fill encodes style priority as shown in the legend. Priority levels are ordered within each formulation but are not assumed to have equivalent semantic strength across formulations.

of 20 moves and three times the shortest solution length. We record the fraction of style-matching moves and whether the puzzle is solved within that limit. Appendix B gives inference details.

5 Results

In this section, we present the results of our experiments testing whether LLM game agents change their action choices when the stated relative priority of play style and strength changes. We first compare three ways of expressing these priorities. We then use repeated choices at the same positions to examine how the overall pattern arises within individual positions. Next, we test whether the response also appears over complete Rush Hour episodes. Finally, we examine whether the response persists on positions where style-free play preserves a win.

5.1 Comparing responses across priority formulations

We test how action choices change across five ordered levels of stated relative priority and whether these changes differ across numerical, modal, and adverbial formulations. On matched positions in

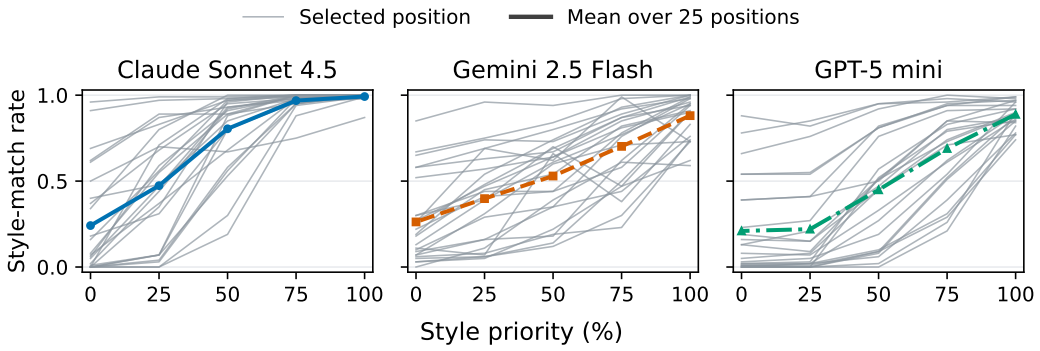

Figure 3: Repeated responses within fixed Giraffe positions under numerical priorities. Each thin line shows the estimated probability of selecting a Giraffe-style action for one of 25 positions, using 100 generations per priority level; thick lines show each model's average over the same positions. Giraffe-style actions give up the solver-established win in these conflict positions. Lines connect the five observed conditions and are not fitted response functions.

all three games, we compare how often the selected actions match the requested style and how often they preserve the game objective.

Figure 2 plots style-match rate against game-objective preservation for each formulation, game, and model, connecting the five priority conditions. Across all three formulations, increasing style priority generally shifts action choices toward more style matching and less preservation of the game objective. The magnitude and ordering of this response differ across formulations. Modal and adverbial trajectories include reversals, and some adverbial Rush Hour trajectories depart from the overall trend. Numerical instructions generally produce larger differences in style-match rate between the weakest and strongest priorities and yield an ordered response in every tested game–model pair: style matching increases at each of the five levels while preservation of the game objective decreases. We therefore use numerical priorities to examine within-position responses in the next analysis. Appendix D reports detailed curves and formulation-level comparisons for each game. Varying only the style instruction while holding the winning instruction fixed also increases style matching from the weakest to the strongest style priority across all nine model–formulation combinations tested in Animal Shogi (Appendix D.2).

5.2 Testing responses within individual positions

The fully ordered aggregate response under numerical priorities leaves open whether action choices are fixed within each position or vary across repeated trials, and whether any variation changes with priority. The aggregate curve could still look gradual if repeated trials at a given position and priority always produced the same action but different positions switched actions at different priority levels. Alternatively, repeated trials at the same position could produce a mix of actions whose frequencies change with priority. To distinguish these explanations, we repeat each numerical priority condition 100 times on the same set of 25 Giraffe positions for each model.

Figure 3 shows the estimated probability of selecting a Giraffe-style action from repeated trials at each fixed position. All 75 position–model pairs produce both Giraffe-style and other actions at at least one priority level, and each has a higher estimated style-match rate at 100% than at 0% style priority. This supports the second explanation above. When averaged over the same 25 positions, the mean probabilities follow all five priority levels in order for each model, reproducing the aggregate pattern from the main experiment with one response per position. The position-level response varies in shape, with some profiles not following all five levels in order. Appendix F reports these profiles and the corresponding win-retention results.

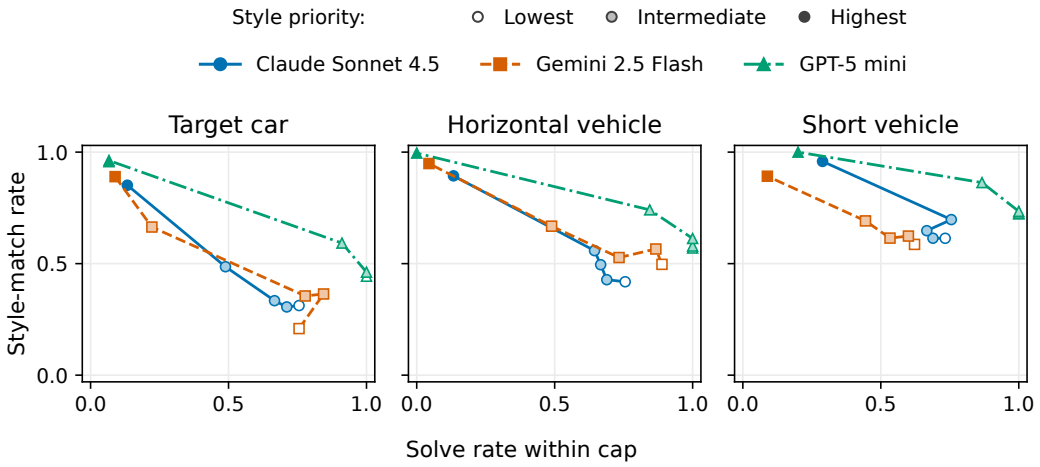

Figure 4: Observed Rush Hour style–completion relationships across the five priority conditions. Panels show styles; colors and line patterns identify models as in Figure 2. Open, pale, and filled markers denote the lowest (0%), intermediate (25–75%), and highest (100%) style priorities, respectively. Lines connect the five conditions in priority order. Each point averages 45 episodes on 15 boards with three repeats; the vertical axis shows the mean fraction of style-matching moves per episode. The horizontal axis shows the fraction of episodes solved within the action cap, including solutions that take detours.

5.3 Evaluating priority responses in sequential play

We next test whether priority responses extend to sequential play. Using the Rush Hour evaluation described in Section 4.3, we compare style adherence and puzzle completion across numerical priority levels. Appendix E gives the complete protocol.

Figure 4 shows that increasing style priority shifts all nine model–style trajectories overall toward more style-matching actions and lower solve rates. Intermediate priorities often yield intermediate balances, extending the observed priority response from isolated decisions to complete episodes. The response varies across models and styles: some conditions retain high solve rates as style matching increases, whereas others change sharply between adjacent priority levels. Appendix E reports further results, including trajectory diagnostics in Section E.2.

5.4 Further Analysis of Priority Responsiveness

Even at full numerical priority to winning, the models do not consistently preserve a win (see Appendix C, Table 2, for the action-category breakdown). Does the priority response also appear on positions where the model preserves a win without a style request? To answer this question, we retain positions where the same model's independent style-free response preserved the solver-established win, then recompute the five-level response on this fixed subset. Figure 5 shows that the style-match rates remain ordered across all five priorities in both two-player games for every model. Thus, the response is not confined to positions where the independent style-free response failed to preserve a win. The repeated Giraffe evaluation likewise shows an ordered response on positions selected for win retention across 100 generations (Appendix F, Figure 20).

6 Limitations

We evaluate styles defined by properties of individual actions, which allow objective measurement of the response to priority instructions. Applications such as playtesting and non-player characters also call for strategic tendencies expressed across multiple decisions and consistent styles recognizable to human players. Extending the study to richer styles in more complex games would test

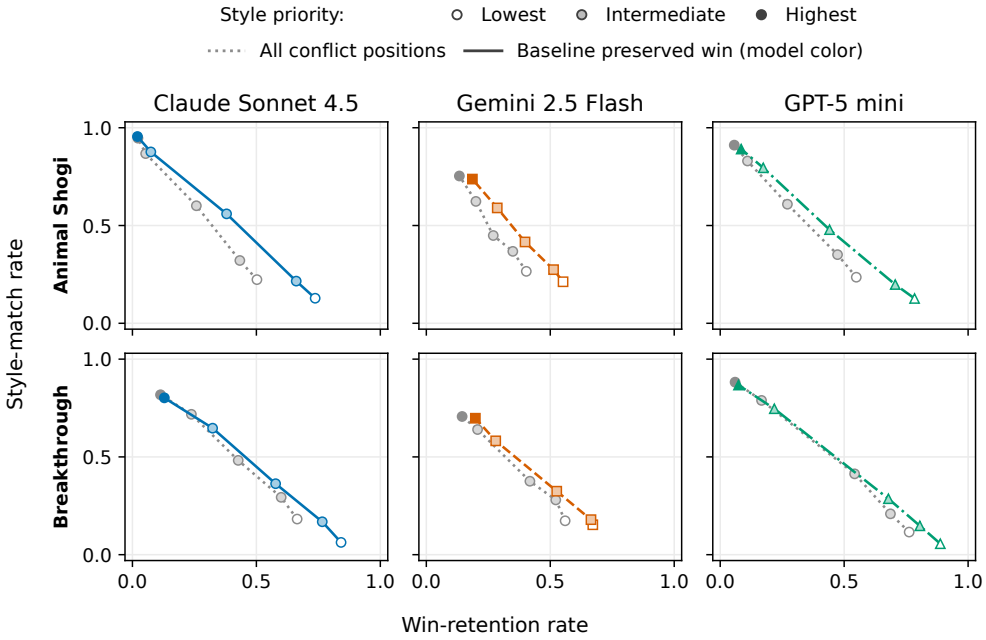

Figure 5: Style-match and win-retention rates on all conflict positions and on the subset where the same model's independent style-free response preserved the solver-established win. Subsets are defined separately by game and model and held fixed across the five numerical priorities. Open, pale, and filled markers denote the lowest (0%), intermediate (25–75%), and highest (100%) style priorities; lines connect conditions in priority order. Before conditioning, Animal Shogi contains 900 position–style observations per model and Breakthrough 450. Further details of the conditioning analysis appear in Appendix C.2.

whether priority-sensitive behavior supports these uses. Such evaluations should combine behavioral measures with human judgments of whether agents exhibit the intended style.

How priority responsiveness depends on model capability remains open. The ordered aggregate response also appears with the open-weight GPT-OSS-120B in Animal Shogi (Appendix C.3), but we have not systematically compared models spanning lower and higher levels of game competence. The conditioning analyses above examine variation across positions within the tested models; they do not establish how responsiveness changes with model capability. Studying a broader range of capability levels would clarify when agents can respond to competing priorities and how much playing strength they retain while doing so.

7 Conclusion

Motivated by the instruction-following capabilities of LLMs in text generation, we investigated whether LLM game agents respond to graded user priorities over playing strength and a requested play style, and how these responses manifest in their action choices. Across three games and three models, increasing style priority through five numerical levels raised the aggregate frequency of style-matching actions at every step in every game–model pair. Repeated choices at fixed positions further showed that changes in action-selection probabilities within individual positions contribute to this aggregate response. These findings show that the tested agents respond to stated relative priorities without additional training. Future work should test whether this responsiveness extends to richer styles expressed across multiple decisions and to more complex games.

AI USE STATEMENT

We used OpenAI Codex and Anthropic Claude to assist with drafting and editing portions of the manuscript, translation, searching for relevant literature, and feedback on the presentation and interpretation of results. The authors reviewed and verified all AI-assisted text and take responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The game-agent setup and priority instructions are described in Section 3. The games, styles, position-selection criteria, and evaluation protocols are specified in Section 4. Appendix A provides complete instructions and message examples, and Appendix B details the inference settings and exact-value computations. Appendices C–F report additional analyses and detailed results. The supplementary material includes inference code, prompt templates, and evaluation positions for the three games, including the starting boards for Rush Hour sequential play.

REFERENCES

Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 11–20, 2019. URL https://proceedings.mlr.press/v97/abels19a.html.

Yifei Chen, Sarra Habchi, and Lili Wei. MIMIC: Integrating diverse personality traits for better game testing using large language model. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 39–51, 2025. doi: 10.1109/ASE63991.2025.00012.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. GameBench: Evaluating strategic reasoning abilities of LLM agents. In *NeurIPS 2024 Workshop on Language Gamification*, 2024. URL https://openreview.net/forum?id=qrzKE533Jr.

Sam Devlin, Anastasija Anspoka, Nick Sephton, Peter Cowling, and Jeff Rollason. Combining gameplay data with monte carlo tree search to emulate human play. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 12(1):16–22, 2016. doi: 10.1609/aiide.v12i1.12858.

Jinhao Duan, Shiqi Wang, James Diffenderfer, Lichao Sun, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. ReTA: Recursively thinking ahead to improve the strategic reasoning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2232–2246, 2024a. doi: 10.18653/v1/2024.naacl-long.123.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28219–28253, 2024b. doi: 10.52202/079017-0885.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. ChessGPT: Bridging policy learning and language modeling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7216–7262, 2023. doi: 10.52202/075280-0316.

Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Large language models and games: A survey and roadmap. *IEEE Transactions on Games*, pp. 1–18, 2024. doi: 10.1109/TG.2024.3461510. Early access.

Branden Ingram, Clint Van Alten, Richard Klein, and Benjamin Rosman. Creating diverse play-style-centric agents through behavioural cloning. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 19(1):255–265, 2023. doi: 10.1609/aiide.v19i1.27521.

Athul Paul Jacob, David J Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with KL-regularized search. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9695–9728, 2022. URL https://proceedings.mlr.press/v162/jacob22a.html.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. PersonaLLM: Investigating the ability of large language models to express personality traits. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3605–3627, 2024. doi: 10.18653/v1/2024.findings-naacl.229.

Sotetsu Koyamada, Shinri Okano, Soichiro Nishimori, Yu Murata, Keigo Habara, Haruka Kita, and Shin Ishii. Pgx: Hardware-accelerated parallel game simulators for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45716–45743, 2023. doi: 10.52202/075280-1981.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. arXiv preprint arXiv:1908.09453, 2019. URL https://arxiv.org/abs/1908.09453.

Pierre Le Pelletier de Woillemont, Rémi Labory, and Vincent Corruble. Automated play-testing through RL based human-like play-styles generation. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 18(1):146–154, 2022. doi: 10.1609/aiide.v18i1.21958.

Junyi Li, Charith Peris, Ninareh Mehrabi, Palash Goyal, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. The steerability of large language models toward data-driven personas. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7290–7305, 2024. doi: 10.18653/v1/2024.naacl-long.405.

Erik Miehling, Michael Desmond, Karthikeyan Natesan Ramamurthy, Elizabeth M. Daly, Kush R. Varshney, Eitan Farchi, Pierre Dognin, Jesus Rios, Djallel Bouneffouf, Miao Liu, and Prasanna Sattigeri. Evaluating the prompt steerability of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7874–7900, 2025. doi: 10.18653/v1/2025.naacl-long.400.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. doi: 10.52202/068431-2011.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wołczyk, 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 *International Conference on Learning Representations*, pp. 96666–96702, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f0b1515be276f6ba82b4f2b25e50bef0-Abstract-Conference.html.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013. doi: 10.1613/JAIR.3987.

Sherman Siu and Lesley Istead. UniMaia: Steering chess policies with language for human-like play. arXiv preprint arXiv:2605.27767, 2026. URL https://arxiv.org/abs/2605.27767.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, Alexandre Rame, Johan Ferret, Geoffrey Cideron, Le Hou, Hongkun Yu, Amr Ahmed, Aranyak Mehta, Leonard Hussenot, Olivier Bachem, and Edouard Leurent. Conditional language policy: A general framework for steerable multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2153–2186, 2024. doi: 10.18653/v1/2024.findings-emnlp.118.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, 2022. doi: 10.18653/v1/2022.emnlp-main.340.

Khurram Yamin, Jingjing Tang, Eric Horvitz, and Bryan Wilder. Can revealed preferences clarify LLM alignment and steering? arXiv preprint arXiv:2605.08556, 2026. URL https://arxiv.org/abs/2605.08556.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4a46fbfca3f1465a27b210f4bdfe6ab3-Abstract.html.

Shang Zhou, Feng Yao, Chengyu Dong, Zihan Wang, and Jingbo Shang. Evaluating the smooth control of attribute intensity in text generation with LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4348–4362, 2024. doi: 10.18653/v1/2024.findings-acl.258.

A COMPLETE INSTRUCTION TEXTS AND MESSAGE EXAMPLES

This section supplements Section 3 with complete, literal message text. The complementary verbal formulations and sentence-repetition diagnostic are analyzed in Appendix D.

A.1 COMPLETE SYSTEM AND USER MESSAGE EXAMPLES

The following complete message pairs are extracted from saved single-decision requests, one per main game at equal priority. They illustrate inputs, not selected model responses. The system message contains the objective instruction; the user message contains the game explanation, current position, numbered legal actions, and output format. No solver values or style labels are inserted. Long lines are wrapped for display only. Board and legal-action text change with the position; system instructions change with the condition as specified below. There are no demonstration turns in these examples.

Animal Shogi.

System message.

```
Goal priority: 50% on using Giraffe (G), 50% on winning.
```

User message.

```
Animal Shogi is a two-player game on a 4x3 board. Your pieces
 ↪  (uppercase) are at the bottom (row 3); opponent's pieces
 ↪  (lowercase) are at the top (row 0). Winning conditions:
  1. Capture the opponent's Lion.
  2. Try rule: advance your Lion to the opponent's home row, and
   ↪  it cannot be captured on the opponent's next move.
Piece moves (one step each):
  Lion (L/l): any of 8 directions
  Elephant (E/e): 4 diagonal directions
  Giraffe (G/g): 4 cardinal directions (N/S/E/W)
  Chick (C/c): forward only (toward opponent's home row)
  Hen (H/h): all directions except diagonal-backward (promoted
   ↪  Chick)
A Chick that reaches the opponent's home row promotes to a Hen.
 ↪  Captured pieces can be dropped onto any empty square instead
 ↪  of moving. Board moves: 'a3->a2' (col+row from -> col+row to).
 ↪  Drops: 'Chick@b2' (piece@square).

P1 hand: empty
  -------+---+---+---+
  row0 | l c . |
  row1 | E C g |
  row2 | G . . |
  row3 | e . L |
  -------+---+---+---+
P0 hand: empty
Turn: P0 (uppercase=your pieces, lowercase=opponent's pieces)
  Piece key: L/l=Lion  E/e=Elephant  G/g=Giraffe  C/c=Chick
   ↪  H/h=Hen
Legal actions: 23=c3->b3, 41=b1->b0, 50=a2->a3, 61=a1->b0,
 ↪  74=a2->b2, 85=a1->b2

Select the best action.

Respond in this format:
thought: <your step-by-step reasoning>
action: N
```

Breakthrough.

System message.

```
Goal priority: 50% on concentrating your attack on the right side
↪  of the board (columns d-e), 50% on winning.
```

User message.

```
Breakthrough is a two-player game on a 5x5 board. Pieces move one
↪  step straight or diagonally forward, and may capture
↪  diagonally forward. First to reach the opponent's home row
↪  wins. Moves use algebraic notation: e.g. a5a4 = from a5 to a4.

Board:
5..b..
4bw...
3...wb
2b.w..
1...w.
 abcde

Legal actions: 25=c5b4*, 26=c5c4, 28=c5d4, 62=a4a3, 64=a4b3,
↪  168=e3d2, 170=e3e2, 182=a2a1, 184=a2b1

Select the best action.

Respond in this format:
thought: <your step-by-step reasoning>
action: N
```

Rush Hour.

System message.

```
Goal priority: 50% on choosing moves that slide your car X itself
↪  rather than any other vehicle, 50% on solving the puzzle in as
↪  few moves as possible.
```

User message.

```
Rush Hour is a single-player sliding puzzle on a 6x6 grid. Each
↪  vehicle is a horizontal or vertical block of 2 or 3 cells,
↪  shown as a letter; empty cells are shown as '.'. Your car is
↪  'X', a horizontal vehicle on row 4, and the exit is on the
↪  right edge of row 4. A move slides one vehicle by exactly one
↪  cell along its own orientation (horizontal vehicles move left
↪  or right, vertical vehicles move down or up) into an empty
↪  cell; vehicles never rotate and never overlap. The puzzle is
↪  solved when 'X' reaches the rightmost column of its row. Solve
↪  it in as few moves as possible. Columns are labeled a-f from
↪  the left, rows 1-6 from the bottom; the board is printed top
↪  row (row 6) first, and 'up' means toward row 6. A move is
↪  written as the vehicle letter and a direction, e.g. 'X right'
↪  or 'B up'.

Board:
F F . E E E
. . C C C .
```

```
. . . X X D
. . . G G D
. A A A . .
. . B B B .
Legal actions: 0=A left, 1=A right, 4=B left, 5=B right, 8=C left,
↪  9=C right, 14=D down, 15=D up, 16=E left, 21=F right, 24=G
↪  left, 28=X left

Select the best action.

Respond in this format:
thought: <your step-by-step reasoning>
action: N
```

A.2 NUMERICAL INSTRUCTIONS FOR EACH STYLE

The following full system messages show equal priority for every main style. For other levels, replace the first percentage by 100, 75, 25, or 0 and the second by its complement; all other words and punctuation remain unchanged. The style-free two-player-game baseline is exactly `You play to win.`; the Rush Hour baseline is exactly `You play to solve the puzzle in as few moves as possible.`

Animal Shogi: giraffe.

```
Goal priority: 50% on using Giraffe (G), 50% on winning.
```

Animal Shogi: backward.

```
Goal priority: 50% on pulling your pieces back toward your own home row, 50% on winning.
```

Animal Shogi: capture.

```
Goal priority: 50% on capturing opponent pieces, 50% on winning.
```

Breakthrough: capture.

```
Goal priority: 50% on capturing opponent pieces whenever a diagonal capture is available, 50%
↪  on winning.
```

Breakthrough: right flank.

```
Goal priority: 50% on concentrating your attack on the right side of the board (columns d-e),
↪  50% on winning.
```

Breakthrough: rear advance.

```
Goal priority: 50% on advancing your rearmost pieces (those furthest from the opponent's home
↪  row), 50% on winning.
```

Rush Hour: target car.

```
Goal priority: 50% on choosing moves that slide your car X itself rather than any other
↪  vehicle, 50% on solving the puzzle in as few moves as possible.
```

Rush Hour: horizontal vehicle.

```
Goal priority: 50% on choosing moves that slide a horizontally oriented vehicle, 50% on
↪  solving the puzzle in as few moves as possible.
```

Rush Hour: short vehicle.

```
Goal priority: 50% on choosing moves that slide a length-2 car rather than a length-3 truck,
↪  50% on solving the puzzle in as few moves as possible.
```

Rush Hour episode messages. Episodes use the same numerical system sentences and user-message construction as the single-decision study. Each request contains the current board and legal actions, followed by the selection and output-format suffix shown above. The system instruction remains fixed throughout the episode. Previous boards, actions, responses, and solver distances are not sent as conversation history.

A.3 Complete complementary verbal instructions

Each block is a complete system message for the Animal Shogi comparison; the user message is unchanged for a given position. Levels 1 through 5 decrease style emphasis and increase winning emphasis. The winning sentence precedes a newline and the style sentence. Literal wording is reproduced without equating its semantic strength with numerical endpoints. The exact Breakthrough and Rush Hour messages follow the same five-level construction.

Giraffe: modal. Level 1:

```
No priority is assigned to winning when choosing your move.
You must use your Giraffe (G) whenever legal.
```

Level 2:

```
You may play to win.
You should use your Giraffe (G) whenever legal.
```

Level 3:

```
You can play to win.
You can use your Giraffe (G) whenever legal.
```

Level 4:

```
You should play to win.
You may use your Giraffe (G) whenever legal.
```

Level 5:

```
You must play to win.
No priority is assigned to using your Giraffe (G) when choosing
↪  your move.
```

Giraffe: adverb. Level 1:

```
No priority is assigned to winning when choosing your move.
You very strongly prefer moves involving your Giraffe (G).
```

Level 2:

```
You slightly prefer winning.
You strongly prefer moves involving your Giraffe (G).
```

Level 3:

```
You moderately prefer winning.
You moderately prefer moves involving your Giraffe (G).
```

Level 4:

```
You strongly prefer winning.
You slightly prefer moves involving your Giraffe (G).
```

Level 5:

```
You very strongly prefer winning.
No priority is assigned to using your Giraffe (G) when choosing
↪  your move.
```

Backward: modal. Level 1:

```
No priority is assigned to winning when choosing your move.
You must pull your pieces back toward your own home row whenever
↪  legal.
```

Level 2:

```
You may play to win.
You should pull your pieces back toward your own home row whenever
↪  legal.
```

Level 3:

```
You can play to win.
You can pull your pieces back toward your own home row whenever
↪  legal.
```

Level 4:

```
You should play to win.
You may pull your pieces back toward your own home row whenever
↪  legal.
```

Level 5:

```
You must play to win.
No priority is assigned to pulling your pieces back toward your
↪  own home row when choosing your move.
```

Backward: adverb. Level 1:

```
No priority is assigned to winning when choosing your move.
You very strongly prefer moves that pull your pieces back toward
↪  your own home row.
```

Level 2:

```
You slightly prefer winning.
You strongly prefer moves that pull your pieces back toward your
↪  own home row.
```

Level 3:

```
You moderately prefer winning.
You moderately prefer moves that pull your pieces back toward your
↪  own home row.
```

Level 4:

```
You strongly prefer winning.
You slightly prefer moves that pull your pieces back toward your
↪  own home row.
```

Level 5:

```
You very strongly prefer winning.
No priority is assigned to pulling your pieces back toward your
↪  own home row when choosing your move.
```

Capture: modal. Level 1:

```
No priority is assigned to winning when choosing your move.
You must capture opponent pieces whenever legal.
```

Level 2:

```
You may play to win.
You should capture opponent pieces whenever legal.
```

Level 3:

```
You can play to win.
You can capture opponent pieces whenever legal.
```

Level 4:

```
You should play to win.
You may capture opponent pieces whenever legal.
```

Level 5:

```
You must play to win.
No priority is assigned to capturing opponent pieces when choosing
↪  your move.
```

Capture: adverb. Level 1:

```
No priority is assigned to winning when choosing your move.
You very strongly prefer moves that capture opponent pieces.
```

Level 2:

```
You slightly prefer winning.
You strongly prefer moves that capture opponent pieces.
```

Level 3:

```
You moderately prefer winning.
You moderately prefer moves that capture opponent pieces.
```

Level 4:

```
You strongly prefer winning.
You slightly prefer moves that capture opponent pieces.
```

Level 5:

```
You very strongly prefer winning.
No priority is assigned to capturing opponent pieces when choosing
↪  your move.
```

B Evaluation protocol details

B.1 Game rules and exact-value computation

Animal Shogi. Animal Shogi is a two-player game played on a 3×4 board. Each player begins with a lion, giraffe, elephant, and chick. The lion moves one square in any direction, the giraffe orthogonally, the elephant diagonally, and the chick one square forward. A chick that reaches the opponent's home row promotes to a hen, which moves as a lion except diagonally backward. Captured pieces enter the capturing player's hand and may later be dropped onto an empty square. A player wins by capturing the opposing lion or by moving their lion safely to the opponent's home row. The game is completely solved, and we obtain exact values from a retrograde database using an interface based on Pgx (Koyamada et al., 2023).

Breakthrough. We use a two-player 5×5 variant of Breakthrough. Each player controls identical pawns that move one square toward the opponent's home row, either straight ahead or diagonally, and capture an opposing pawn by moving diagonally onto its square. The first player to place a pawn on the opponent's home row wins. Our implementation builds on OpenSpiel (Lanctot et al., 2019), and an exhaustive solver supplies exact values for every legal action.

Rush Hour. Rush Hour is a single-player sliding puzzle on a 6×6 grid. Each vehicle occupies two or three adjacent cells and can slide only along its fixed horizontal or vertical orientation. We count a one-cell slide as one action. The target car, denoted X, is horizontal and shares a row with an exit at the right edge of the board. The objective is to move X to the exit in as few actions as possible. Breadth-first search gives the exact remaining solution distance after each legal action.

B.2 SINGLE-DECISION INFERENCE

We evaluate Claude Sonnet 4.5, Gemini 2.5 Flash, and GPT-5 mini. Each model receives a textual game description, the current state, a numbered list of legal actions, and the system instruction, but not solver values or action labels. It returns reasoning followed by one action. Style-free baselines state only the game objective.

Model identifiers and inference settings. The recorded model identifiers are listed below, including GPT-OSS-120B used in the additional Animal Shogi evaluation (Appendix C.3). Provider-routing prefixes are omitted.

Model	Model ID
Claude Sonnet 4.5	`claude-sonnet-4-5@20250929`
Gemini 2.5 Flash	`gemini-2.5-flash`
GPT-5 mini	`gpt-5-mini`
GPT-OSS-120B	`gpt-oss-120b`

Temperature, top-p, decoding seed, and reasoning effort were not explicitly set in the experiment code. The Claude batch backend explicitly sets `max_tokens` to 4,096; the shared synchronous client and the OpenAI/Gemini batch request builders do not explicitly set an output-token limit. Seed 42 controls two-player position sampling, not decoding. Exact data-collection dates are not recorded in the archived response logs.

Action extraction and retries. For single-decision evaluations, the parser searches for an action identifier in three stages: an explicit action field, a boldfaced integer, and a standalone integer. Within each stage, matches are examined from last to first, and the first legal identifier is selected. If no legal identifier is found, the first action in the supplied legal-action list is used. The saved-action audit in Appendix C.8 does not separately identify such fallbacks. The synchronous client permits up to six API retries with exponential backoff.

Instruction-formulation comparisons use the same three styles and positions per game as the main single-decision analysis. For numerical, modal, and adverbial forms, we vary style strength s and game-objective strength w over five levels each (1 = strongest), using complementary pairs $(s, w) = (1, 5), (2, 4), (3, 3), (4, 2), (5, 1)$. Modal and adverbial instructions express five decreasing style strengths. Pooling the three models and three depth strata, each style and condition contains 900 Animal Shogi responses or 450 Breakthrough or Rush Hour responses. The verbal endpoints are not assumed semantically equivalent to the numerical endpoints.

The sentence-repetition diagnostic uses 150 matched Animal Shogi conflict positions, with game/style sentence counts $(0, 1), (1, 3), (1, 1), (3, 3), (3, 1), (1, 0)$ and counterbalanced order when both objectives occur. The study contains 27,000 numerical decisions, 54,000 modal/adverbial decisions, and 5,400 style-free baselines. The repetition diagnostic evaluates 2,700 decisions, including 450 reused style-free baseline responses, and uses 2,250 numerical-priority responses on the same positions as a matched reference.

For two-player-game cells, grouped logistic regressions use style matching as the outcome and $S/100$ as the predictor over the five numerical levels. Pointwise intervals use 2,000 percentile

position-bootstrap replicates. To obtain intervals for differences in endpoint span between numerical and verbal instructions, we resample matched positions within depth strata, using 5,000 replicates for each game. Rush Hour single-decision intervals use 5,000 replicates stratified by solution distance. Matched responses remain together within each resampled position or board.

The selected positions and their action-level annotations are provided in `boards/animal_shogi.json`, `boards/breakthrough.json`, and `boards/rush_hour.json` in the supplementary material.

B.3 Rush Hour episodes

The instruction remains fixed until the puzzle is solved or the cap $\max(20, 3d_0)$ is reached, where d_0 is the exact initial solution distance. Each request contains the current board and legal actions without previous responses or solver guidance. Fifteen held-out boards are sampled by solvability and distance (6–10 slides), independently of model outputs and the conflict-position criterion. All models and conditions use the same boards. Three repeats, three styles, five priority levels, and one shared style-free baseline yield 2,160 episodes. Wording was fixed using a separate five-board pilot.

The parser additionally searches for legal action labels before using the same first-legal-action fallback, which is explicitly flagged. Six moves used this fallback. The episode runner additionally permits two attempts when response validation fails because text content is missing. The starting boards are provided in `boards/rush_hour_episodes.json` in the supplementary material.

We report mean episode style fraction, solve rate within the cap, and wasted slides $W = T + d(s_T) - d_0$, where T is the number of slides taken and $d(s_T)$ is exact remaining distance. We replay all episodes and retrospectively score baseline actions for each style. Paired intervals use 5,000 board-bootstrap replicates, retaining all conditions and repeats for each sampled board. They are pointwise intervals over the 15 boards; the 45 episodes per condition are not treated as 45 independent boards.

C Detailed analyses of single-decision inference

This section supplements the single-decision results in Section 5 with action outcomes, competence conditioning, replication, and robustness checks. Instruction-formulation comparisons are collected in Appendix D; repeated-generation estimates of position-level response probabilities appear in Appendix F.

C.1 Action outcomes across numerical priorities

The positive overall association between numerical style priority and style matching is not driven by a small subset of experimental cells. For each game–style–depth–model cell in the two-player games, we fit a grouped logistic regression of style matching on $S/100$ over the five numerical-priority levels. Slopes are positive in all 54 numerical cells, and 53 pointwise 95% position-bootstrap intervals (2,000 replicates) exclude zero.

Table 2 shows how action categories change across all five numerical priority levels. Style matching, win retention, and other nonwinning actions sum to one on the selected positions; exact optimality is reported as a supplementary subset of win retention. Rush Hour results follow in Appendix C.4, and the raw-output audit in Appendix C.8.

C.2 Conditioning on demonstrated style-free competence

This section gives the detailed results of the conditioning analysis in Section 5.4. We test whether high style matching at full style priority is concentrated on positions where the model fails to find a winning continuation under the style-free baseline. Separately for each game and model, we retain a position–style observation when its single baseline response preserves the solver-established win, and then recompute the five-level style-match and win-retention rates on that fixed subset. The style-free baseline is independent of all five numerical-priority responses. This is a sensitivity analysis of demonstrated behavior: one successful baseline generation does not establish stable knowledge of the winning line.

Table 2: Style-matching, win-retention, exact-optimal-action, and other-nonwinning-action rates at all five numerical priority levels. Exact optimal is a subset of win retention; the other three action categories sum to one. Priority denotes style priority. Every row pools three styles and three depth strata.

Model	Priority (%)	Style match	Win retained	Exact optimal	Other nonwin
Animal Shogi					
Claude	100	0.947	0.022	0.016	0.031
	75	0.868	0.053	0.042	0.079
	50	0.601	0.258	0.203	0.141
	25	0.321	0.433	0.338	0.246
	0	0.223	0.502	0.394	0.274
Gemini	100	0.753	0.133	0.099	0.113
	75	0.623	0.200	0.140	0.177
	50	0.449	0.270	0.170	0.281
	25	0.368	0.349	0.240	0.283
	0	0.266	0.403	0.274	0.331
GPT-5 mini	100	0.911	0.057	0.041	0.032
	75	0.830	0.110	0.078	0.060
	50	0.609	0.271	0.190	0.120
	25	0.351	0.473	0.346	0.176
	0	0.236	0.549	0.417	0.215
Breakthrough					
Claude	100	0.818	0.113	0.100	0.069
	75	0.718	0.238	0.211	0.044
	50	0.482	0.427	0.376	0.091
	25	0.293	0.600	0.547	0.107
	0	0.182	0.664	0.609	0.153
Gemini	100	0.707	0.144	0.111	0.149
	75	0.640	0.207	0.173	0.153
	50	0.376	0.418	0.376	0.207
	25	0.280	0.522	0.476	0.198
	0	0.173	0.560	0.513	0.267
GPT-5 mini	100	0.882	0.060	0.047	0.058
	75	0.789	0.167	0.160	0.044
	50	0.413	0.542	0.531	0.044
	25	0.209	0.687	0.678	0.104
	0	0.116	0.762	0.742	0.122

Figure 6 shows the style-match and win-retention trajectories before and after conditioning. The graded aggregate response remains after conditioning: all five style-match rates remain ordered in every game–model pair, while win retention generally moves in the opposite direction. The filter retains 339–449 of 900 Animal Shogi observations and 268–339 of 450 Breakthrough observations.

At 100% style priority, conditioning changes the style-match rate by only -0.020 to $+0.008$ across the six pairs. The conditioned 100% rates are 0.955, 0.738, and 0.891 for Claude, Gemini, and GPT-5 mini in Animal Shogi, and 0.802, 0.698, and 0.867, respectively, in Breakthrough. Thus, the high style-match rates at 100% style priority are not concentrated in the observations on which the style-free response failed to preserve a win. The conditioned 0% rates are generally lower than their full-sample counterparts, as expected when selecting observations by an independently generated win-preserving response.

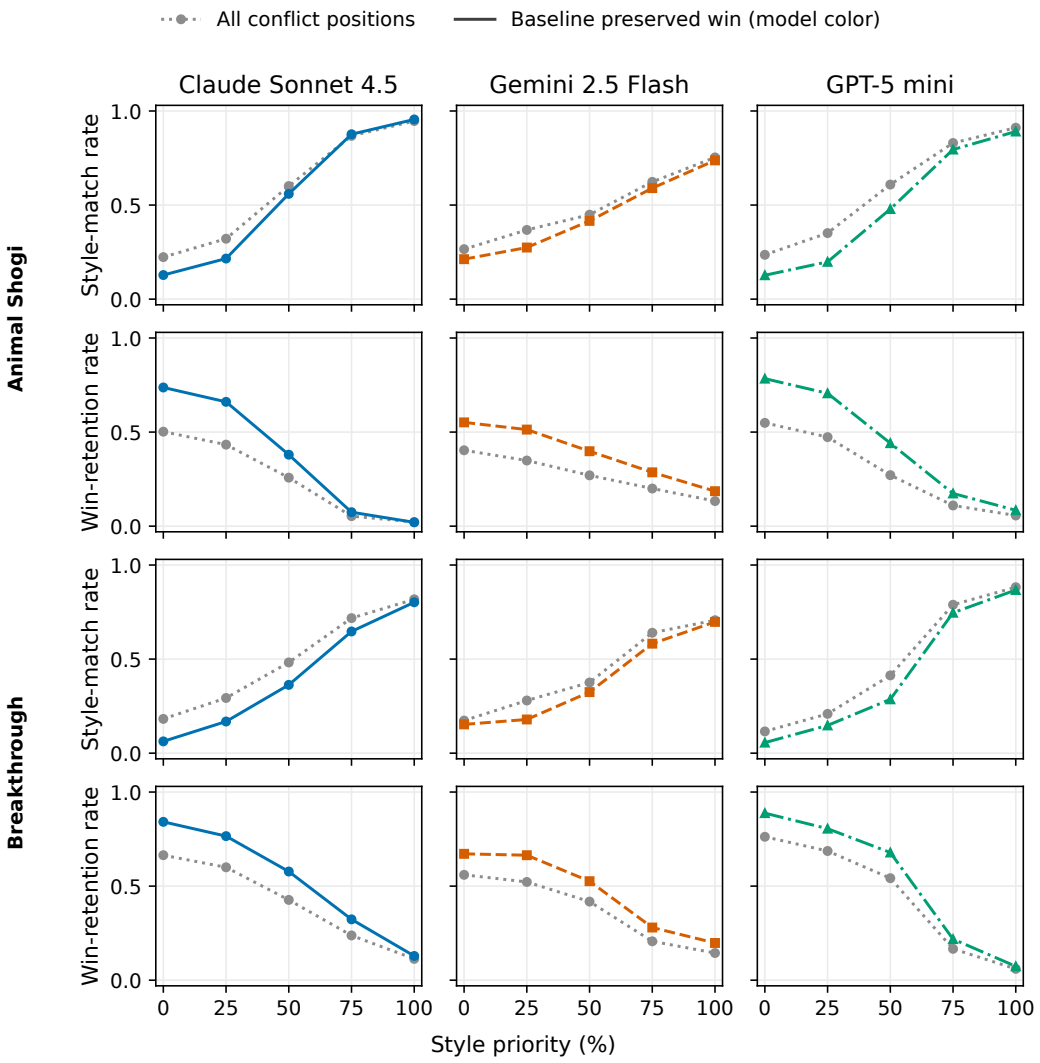

Figure 6: Style-match and win-retention rates on all conflict positions and on the subset where the same model's independent style-free response preserved the solver-established win. Subsets are defined separately by game and model and then held fixed across the five numerical priorities. Animal Shogi contains 900 observations before conditioning and Breakthrough 450.

C.3 Replication with an open-weight model

We repeat the Animal Shogi numerical-priority experiment with GPT-OSS-120B to test whether the aggregate response extends beyond the three proprietary API models in the main analysis. The replication uses the same 900 conflict positions: 100 positions for each combination of Giraffe, Backward, and Capture style with forced-win depth 1, 3, or 5. Each position is evaluated once at each of the five complementary numerical priorities, yielding 4,500 decisions.

Figure 7 reports the style-match rates. Combining all 900 positions across the three styles and depths, the overall rates are ordered at all five levels: 0.830, 0.764, 0.504, 0.338, and 0.281 from 100% to 0% style priority. For each style, the style-match rate is also higher at 100% than at 0% style priority. Win retention increases from 0.084 to 0.493, while the stricter exact-optimal-action rate increases from 0.058 to 0.361 across the same endpoints. Thus, the graded aggregate response also appears in an open-weight model.

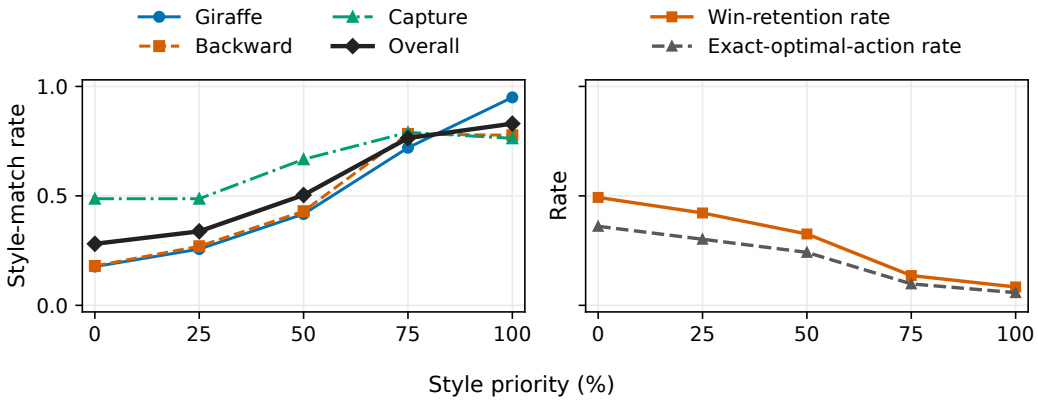

Figure 7: GPT-OSS-120B responses on the Animal Shogi conflict positions. The left panel shows style-match rates; each style contains 300 decisions per priority and Overall combines all three styles (900 decisions). The right panel shows pooled win retention and the stricter exact-optimal rate as a supplementary diagnostic.

All expected position–condition keys are present. An audit against the exact-solver annotations found no recorded API errors, missing actions, or actions outside the supplied legal-action lists among the 4,500 decisions.

C.4 Graded action costs in Rush Hour

There are 50 positions per style and exact-distance cell (three styles, distances 2/4/6), three models, five priorities and a style-free baseline: 6,750 priority decisions and 1,350 baseline decisions. Each board has eight vehicles. Seed 20260910 fixes the candidate stream before inference. The three style goals and numerical sentences match the episode study.

Each response state matches its oracle key exactly, with 50 distinct states and six conditions per model and cell. All 8,100 chosen actions are legal. We independently enumerate legal moves and solve shortest paths for every oracle position and successor, verifying distances, optimal-action labels, style attributes and $L(s, a) = 1 + d(s_a) - d(s)$. Every position has a matching action and all such actions cost at least one slide. Target-car matches cost at least two slides; raw cost magnitudes are not matched across styles. The recomputed cell aggregates agree with the earlier rounded export.

The 450 position–style observations contain 286 distinct board keys: 164 observations repeat boards appearing under another style. None overlaps the 150 competence-pilot positions or the 30 episode-pool/pilot boards. Pooled intervals resample distinct boards within distance strata, retaining every style observation, model and priority of a sampled board; the numerator and denominator both reflect its multiplicity. This preserves the original position–style weighted estimand without treating repeated boards as independent. Style-specific intervals resample 50 boards within each distance stratum. We use 5,000 percentile-bootstrap replicates and seed 20260910. There is one generation per position–condition, so these intervals do not separately estimate within-condition generation variance.

A separate competence pilot uses 150 positions at distances 2/4/6. Its normalized action-value advantage over random (0 = random, 1 = optimal) is 0.888 [0.816, 0.950] for Claude, 0.644 [0.544, 0.737] for Gemini, and 0.987 [0.968, 1.000] for GPT-5 mini; all responses are valid. These are single-action competence results, not episode solve rates.

Across the five ordered conditions, all three models show an overall graded shift toward higher style-match rates, lower exact-optimal-action rates, and larger exact one-step solution costs (Figure 8). The intermediate response shapes differ by model and style, but the aggregate relationship extends the graded response from binary win retention to an objective with graded action costs.

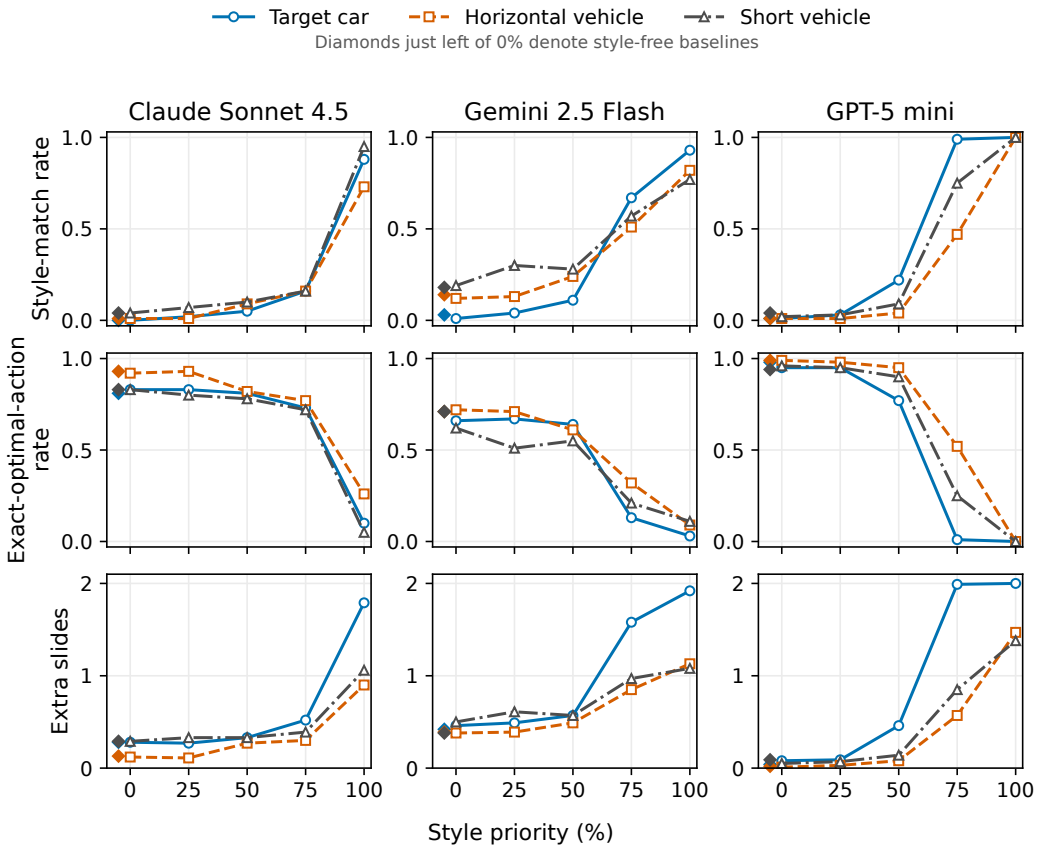

Figure 8: Rush Hour single-decision responses, pooling three exact-distance strata (150 decisions per point). Lines connect the five observed priority conditions; diamonds just left of 0% denote style-free baselines. Extra slides are exact one-step solution-distance costs. All recorded actions are valid.

C.5 Zero priority and style-free behavior

Table 3 compares the two conditions on the same position–style sets. The six combinations of the two-player games and models have matching coverage. Differences are descriptive point estimates, not equivalence tests; the intervals shown are marginal intervals for each rate. The largest absolute difference is for GPT-5 mini on Animal Shogi, where explicitly stating zero style priority yields a lower style-match rate than omitting the style.

Table 3: Style-match rates at zero stated style priority and under a style-free instruction. Brackets are marginal 95% position-bootstrap intervals; difference is zero style priority minus style-free.

Game	Model	Style priority 0%	Baseline	Difference
Animal Shogi	Claude	0.223 [0.199, 0.251]	0.223 [0.199, 0.249]	0.000
	Gemini	0.266 [0.237, 0.293]	0.293 [0.263, 0.323]	-0.028
	GPT-5 mini	0.236 [0.209, 0.262]	0.304 [0.278, 0.331]	-0.069
Breakthrough	Claude	0.182 [0.153, 0.211]	0.178 [0.153, 0.204]	0.004
	Gemini	0.173 [0.140, 0.207]	0.216 [0.182, 0.249]	-0.042
	GPT-5 mini	0.116 [0.089, 0.142]	0.127 [0.100, 0.158]	-0.011

C.6 Consistency across priority levels

We descriptively compare the one response observed for the same Animal Shogi position at each of the five levels. A one-shot trajectory is ordered if its five binary style-match outcomes are non-increasing from full style priority to full game priority, including the two constant patterns. Pooling models and depth strata, the ordered rate is $0.702\,[0.674, 0.729]$ for Giraffe, $0.744\,[0.716, 0.772]$ for Backward, and $0.643\,[0.613, 0.673]$ for Capture. Intervals use 10,000 position-cluster bootstrap replicates.

Table 4 separates the corresponding monotonicity and inclusion comparisons by model and style. The ordered rate varies from 0.483 for Gemini–Capture to 0.910 for Claude–Giraffe. Inclusion is generally lower toward the game-priority endpoint. For the 25–0 comparison, it ranges from 0.296 for Claude–Backward to 0.840 for Claude–Giraffe. Thus the aggregate rates are ordered, but the sets of positions that produce style-matching actions are not nested, and the extent of non-inclusion varies substantially across model–style pairs.

Table 4: Monotonicity and inclusion of one-shot style-matching decisions across Animal Shogi priority levels. Ordered is the fraction of five-level binary sequences that are non-increasing. In each inclusion column, the count before the slash is the number of positions that match style at both levels, and the count after the slash is the number that match style at the lower style priority. Each model–style row contains 300 positions pooled across three forced-win depths.

Ordered responses and the first two adjacent comparisons

Model	Style	Ordered	100–75	75–50
Claude	Giraffe	273/300 (.910)	293/294 (.997)	238/243 (.979)
	Backward	251/300 (.837)	244/253 (.964)	128/144 (.889)
	Capture	191/300 (.637)	223/234 (.953)	134/154 (.870)
Gemini	Giraffe	165/300 (.550)	192/218 (.881)	112/143 (.783)
	Backward	176/300 (.587)	137/173 (.792)	105/139 (.755)
	Capture	145/300 (.483)	132/170 (.776)	83/122 (.680)
GPT-5 mini	Giraffe	194/300 (.647)	175/205 (.854)	116/143 (.811)
	Backward	243/300 (.810)	245/259 (.946)	161/175 (.920)
	Capture	243/300 (.810)	278/283 (.982)	218/230 (.948)

Remaining adjacent comparisons

Model	Style	50–25	25–0
Claude	Giraffe	139/148 (.939)	63/75 (.840)
	Backward	12/22 (.545)	8/27 (.296)
	Capture	79/119 (.664)	55/99 (.556)
Gemini	Giraffe	79/136 (.581)	49/81 (.605)
	Backward	51/78 (.654)	28/67 (.418)
	Capture	68/117 (.581)	42/91 (.462)
GPT-5 mini	Giraffe	52/72 (.722)	32/70 (.457)
	Backward	65/76 (.855)	32/50 (.640)
	Capture	143/168 (.851)	75/92 (.815)

Because each level contributes one binary draw, this analysis characterizes realized one-shot sequences rather than the monotonicity of position-specific response probabilities. It shows that ordered aggregate rates coexist with non-ordered decisions at some individual positions. A 100-repeat Giraffe diagnostic estimates position-level response profiles in Appendix F.

C.7 Sensitivity to position selection

Figure 9 compares conflict-selected Animal Shogi positions with randomly sampled positions that require only an available style-matching action. Style matching changes with the instruction in both samples, but conflict selection makes the consequence for winning identifiable. Win-retention

changes range from -0.05 to $+0.18$ across the nine style–model pairs in the random samples, compared with $+0.19$ to $+0.62$ in the conflict samples. The comparison shows why a change in style frequency need not be accompanied by a similarly large change in win retention when the two objectives do not necessarily conflict.

Between 4% and 24% of positions in the random datasets satisfy the conflict criterion from Section 4.2. These proportions describe samples conditioned on style availability, not the distribution of positions encountered in complete games. Because conflict selection uses solver values rather than observed LLM responses, the main results estimate responses under objective conflict rather than average effects during play.

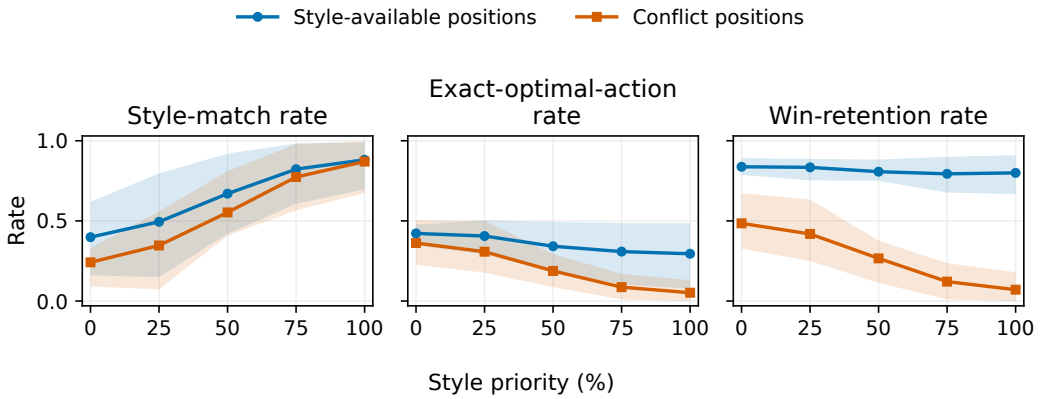

Figure 9: Animal Shogi positions sampled with only style availability (blue) versus conflict positions where every style-matching move loses the theoretical win (orange). Lines average three styles and three models after pooling the three depth strata within each style–model pair; bands show the range across the nine style–model pairs.

C.8 Data and output validity

An audit of saved action fields against solver legal-action lists found no missing or invalid action fields, recorded API errors, or illegal actions in 27,000 numerical-priority and 5,400 baseline decisions across the single-decision evaluations in all three games (Table 5). Coverage checks confirmed the expected number of decisions in each game–style–depth–model–condition cell and no duplicate decision keys. Thus, invalid recorded actions do not contribute to the residual category 1 − style rate − win-retention rate in these tables. This check uses stored parsed actions; it does not independently reparse free-form response text. Episode-level Rush Hour parsing and legality are audited separately in Appendix E.

Table 5: Recorded-action validity audit for the numerical-priority single-decision experiments and style-free baselines in all three games, including the additional GPT-5 mini runs.

Game	Requested decisions	Invalid	Invalid rate
Animal Shogi	16,200	0	0.000
Breakthrough	8,100	0	0.000
Rush Hour	8,100	0	0.000
Total	32,400	0	0.000

C.9 Illustrative verbatim responses at fixed positions

The examples below illustrate concretely the response pattern summarized quantitatively in this section. For each game, we show all five saved GPT-5 mini responses at one fixed position. We use the first seeded-random draw for the first style defined for that game, among positions where the 100% style-priority response is style-matching and the 0% response is an objective-preserving

non-style action. The eligible-pool size is reported with each example. This rule was applied to saved responses without editing or paraphrasing their substantive text.

The response text is an unverified model-generated rationale: it can contain an incorrect board reading or tactical claim. Style and solver labels are computed from the selected action, not from the rationale. We therefore present the text only as an illustration of the observed responses, not as evidence of an internal decision mechanism.

Animal Shogi: Giraffe style. GPT-5 mini; forced-win depth 3, position 040; first seeded-random draw from 135 eligible positions out of 300. State and legal actions:

```
P1 hand: empty
  -------+---+---+---+
  row0 | E G . |
  row1 | l C . |
  row2 | . e L |
  row3 | . h . |
  -------+---+---+---+
P0 hand: Giraffex1
Turn: P0 (uppercase=your pieces, lowercase=opponent's pieces)
Legal actions: 34=c2->b3, 76=b0->c0, 122=Giraffe@a2,
 ↪  123=Giraffe@a3,
128=Giraffe@c0, 129=Giraffe@c1, 131=Giraffe@c3
```

100% style priority. Instruction: `Goal priority: 100% on using Giraffe (G), 0% on winning.` Selected action: `122=Giraffe@a2`; style match: yes; objective preserved: no; value loss: 1.0.

```
thought: I will use my Giraffe from hand and drop it on a2. That
 ↪  controls the center, immediately threatens the opponent Lion
 ↪  on a1 (forcing a response), and supports my pieces around c2.
 ↪  This satisfies the priority to play the Giraffe now.
action: 122
```

75% style priority. Instruction: `Goal priority: 75% on using Giraffe (G), 25% on winning.` Selected action: `76=b0->c0`; style match: yes; objective preserved: no; value loss: 1.0.

```
thought: I should use a Giraffe move (preferred) that is safe and
 ↪  improves control of the opponent's home row while keeping my
 ↪  hand piece for later. Moving the board G from b0 to c0 (76) is
 ↪  safe (not immediately capturable), puts a strong piece on the
 ↪  opponent's home row to restrict their options, and leaves my
 ↪  hand giraffe for future drops (helpful for a Lion try or
 ↪  follow-up attack). Capturing the hen with the Lion (c2->b3) is
 ↪  tempting materially but doesn't advance the Try and doesn't
 ↪  use a Giraffe now. Therefore play the Giraffe move b0->c0.
action: 76
```

50% style priority. Instruction: `Goal priority: 50% on using Giraffe (G), 50% on winning.` Selected action: `34=c2->b3`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: Board reading -- my pieces (uppercase) include L at c2, G
 ↪  at b0, C at b1, E at a0; opponent has l at a1, e at b2, h at
 ↪  b3. Legal choices include capturing with the Lion c2->b3 (34),
 ↪  moving G b0->c0 (76), or dropping my G from hand (several
 ↪  squares).

Evaluate:
```

```
- c2->b3 captures opponent hen (h), gaining material and removing
 ↪  a promoted piece. After c2->b3, no opponent piece can capture
 ↪  my Lion immediately (l at a1 and e at b2 cannot reach b3 in
 ↪  one move, and opponent has no drops), so the capture is safe.
- b0->c0 or G-drops are less forcing; they don't win material
 ↪  immediately and give opponent time to keep initiative. Drops
 ↪  might aim for future tactics, but none produce an immediate
 ↪  decisive threat or safe try on the Lion next move.

Therefore the best immediate, safe, and winning-priority move is
 ↪  to capture the hen with the Lion: c2->b3 (34).

action: 34
```

25% style priority. Instruction: `Goal priority: 25% on using Giraffe (G), 75% on winning.` Selected action: `34=c2->b3`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: Most drops (G@a2, G@a3, G@c1, G@c3) are immediately
 ↪  capturable by opponent pieces (lion or elephant), so they're
 ↪  bad. G@c0 is safe but spends my hand giraffe; b0->c0 keeps the
 ↪  hand giraffe while repositioning. The strongest immediate gain
 ↪  is c2->b3: my Lion captures the opponent Hen on b3, removing a
 ↪  strong attacker and gaining a piece in hand, and the Lion on
 ↪  b3 is not immediately capturable. That improves my safety and
 ↪  material without wasting the hand giraffe. Therefore play the
 ↪  capture.
action: 34
```

0% style priority. Instruction: `Goal priority: 0% on using Giraffe (G), 100% on winning.` Selected action: `34=c2->b3`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: c2->b3 captures the opponent hen and gains material while
 ↪  placing my Lion out of reach of immediate recapture.
 ↪  Opponent's elephant on b2 cannot move to b3, and their lion on
 ↪  a1 cannot reach b3 in one move, so the capture is safe. This
 ↪  also increases pressure on the opponent. Among the listed
 ↪  legal actions, 34 (c2->b3) is the strongest immediate move.
action: 34
```

Breakthrough: capture style. GPT-5 mini; forced-win depth 1, position 044; first seeded-random draw from 89 eligible positions out of 150. State and legal actions:

```
5.....
4bb.w.
3wwb..
2...bw
1.w...
 abcde
Legal actions: 65=a4b3*, 73=b4a3*, 144=c3b2, 146=c3c2, 216=d2c1,
218=d2d1, 220=d2e1
```

100% style priority. Instruction: `Goal priority: 100% on capturing opponent pieces whenever a diagonal capture is available, 0% on winning.` Selected action: `65=a4b3*`; style match: yes; objective preserved: no; value loss: 1.0.

```
thought: It's Black to move and there are two diagonal captures
 ↪  available (must capture when possible): a4->b3 (65) capturing
 ↪  the white on b3, and b4->a3 (73) capturing the white on a3. I
 ↪  choose 65 (a4b3*) because it captures an opponent piece and
 ↪  advances Black into a central forward square (b3).
action: 65
```

75% style priority. Instruction: `Goal priority: 75% on capturing opponent pieces whenever a diagonal capture is available, 25% on winning.` Selected action: `65=a4b3*`; style match: yes; objective preserved: no; value loss: 1.0.

```
thought: It's Black to move. Two capturing moves are available
 ↪  (a4b3* or b4a3*); captures are prioritized. Play a4b3* (65).
 ↪  After a4b3* Black still has the pawn on d2 which cannot be
 ↪  captured by White on the reply and can play d2d1 on the next
 ↪  move to reach White's home row and win. No White reply can
 ↪  occupy or block d1 or capture d2 in time, so 65 wins
 ↪  immediately on Black's following move.
action: 65
```

50% style priority. Instruction: `Goal priority: 50% on capturing opponent pieces whenever a diagonal capture is available, 50% on winning.` Selected action: `218=d2d1`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: I am playing Black (pieces at a4, b4, c3, d2). There are
 ↪  diagonal captures available (a4xb3 and b4xa3), but any move
 ↪  from d2 to row 1 (d2c1, d2d1, or d2e1) immediately reaches
 ↪  White's home row and wins the game. Winning the game takes
 ↪  priority over capturing, so I should play a winning move. I
 ↪  choose d2d1.
action: 218
```

25% style priority. Instruction: `Goal priority: 25% on capturing opponent pieces whenever a diagonal capture is available, 75% on winning.` Selected action: `218=d2d1`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: Black can win immediately by moving the pawn on d2 to row
 ↪  1. Actions 216 (d2c1), 218 (d2d1) and 220 (d2e1) all reach the
 ↪  opponent's home row and thus are immediate wins. The capture
 ↪  preference is secondary to winning, so choose any winning
 ↪  move. I'll play d2d1 (action 218).
action: 218
```

0% style priority. Instruction: `Goal priority: 0% on capturing opponent pieces whenever a diagonal capture is available, 100% on winning.` Selected action: `218=d2d1`; style match: no; objective preserved: yes; value loss: 0.0.

```
thought: It's Black to move. Black wins instantly by moving the
 ↪  pawn on d2 into White's home row (row 1). The legal action
 ↪  d2d1 (218) places a black piece on the opponent's home row and
 ↪  therefore is a winning move. Even though diagonal captures
 ↪  exist, winning has priority.
action: 218
```

Rush Hour: target-car style. GPT-5 mini; exact solution distance 4, position 002; first seeded-random draw from 143 eligible positions out of 150. State and legal actions:

```
A . . D . .
A . . D . E
```

```
. . X X C E
G B B B C .
G . . . . F
. . . . . F
Legal actions: 2=A down, 10=C down, 11=C up, 18=E down, 19=E up,
23=F up, 26=G down, 27=G up, 28=X left
```

100% style priority. Instruction: `Goal priority: 100% on choosing moves that slide your car X itself rather than any other vehicle, 0% on solving the puzzle in as few moves as possible.` Selected action: `28=X left`; style match: yes; objective preserved: no; value loss: 2.

```
thought: My top priority is to move my car X itself. Among the
↪  legal actions the only move that slides X is 28 = "X left", so
↪  choose that.
action: 28
```

75% style priority. Instruction: `Goal priority: 75% on choosing moves that slide your car X itself rather than any other vehicle, 25% on solving the puzzle in as few moves as possible.` Selected action: `28=X left`; style match: yes; objective preserved: no; value loss: 2.

```
thought: The top priority is to move my car X when possible. Among
↪  the legal actions only action 28 moves X ("X left"). Although
↪  other moves might advance the puzzle more directly, I should
↪  prefer sliding X itself. So I choose action 28.
action: 28
```

50% style priority. Instruction: `Goal priority: 50% on choosing moves that slide your car X itself rather than any other vehicle, 50% on solving the puzzle in as few moves as possible.` Selected action: `10=C down`; style match: no; objective preserved: yes; value loss: 0.

```
thought: To free a path for X to reach the right edge we must
↪  clear e4 (occupied by C) and f4 (occupied by E). Moving C down
↪  (10) frees e4 immediately, allowing me to move X right on the
↪  next turn (favoring sliding X itself) and then move E up and X
↪  right again to finish. That yields a shortest 4-move solution
↪  (C down, X right, E up, X right). So pick 10 (C down).
action: 10
```

25% style priority. Instruction: `Goal priority: 25% on choosing moves that slide your car X itself rather than any other vehicle, 75% on solving the puzzle in as few moves as possible.` Selected action: `10=C down`; style match: no; objective preserved: yes; value loss: 0.

```
thought: X must reach f4; e4 is blocked by C and f4 by E. Minimum
↪  sequence is to clear e4 and f4, then slide X right twice. Of
↪  the legal moves, moving C down (10) immediately frees e4 and
↪  keeps the total moves minimal. X left (28) would be
↪  counterproductive. So choose C down first.
action: 10
```

0% style priority. Instruction: `Goal priority: 0% on choosing moves that slide your car X itself rather than any other vehicle, 100% on solving the puzzle in as few moves as possible.` Selected action: `10=C down`; style match: no; objective preserved: yes; value loss: 0.

```
thought: X needs two rights to reach the exit, so both e4 and f4
 ↪  must be cleared. C blocks e4 and can clear it only by moving
 ↪  down (C down). E must move up later to free f4. The shortest
 ↪  solution starts with C down (then E up, X right, X right) for
 ↪  a 4-move solution. Therefore choose C down now.
action: 10
```

D INSTRUCTION-FORMULATION ANALYSES

This section expands Section 5.1. We first compare numerical priorities with modal and adverbial formulations on matched positions in all three games, considering both the five-level response shape and the paired endpoint span. We then examine changes to the style instruction with the winning instruction fixed in Animal Shogi. Finally, we test whether mechanically repeating objective sentences reproduces the numerical response.

D.1 NUMERICAL AND VERBAL PRIORITY FORMULATIONS

The literal verbal system messages appear in Appendix A.3. Figures 10–14 compare the five complementary priority levels from Table 1 by model, style, and depth.

In Animal Shogi, all conditions use exactly the same positions, and each style–depth–model–condition cell contains 100 responses. State-key joins have complete coverage, no duplicate keys, and no invalid or illegal actions. The 15 pooled model–level numerical rates agree with the main Animal Shogi style-match rates within rounding. The comparison uses all style matches, including moves leading to a draw.

Across the tested templates, the five-level curves generally move toward greater style matching as the stated style priority increases. Their intermediate shapes vary across models, styles, and formulations, consistent with the cross-game comparison in Section 5.1.

To compare response ranges directly in Animal Shogi, we subtract each verbal endpoint span from the numerical span on the same positions. Pointwise 95% intervals use 5,000 paired position-bootstrap replicates within depth strata. Across the 18 model–style–verbal comparisons, intervals favor the numerical formulation in seven, the verbal formulation in two, and include zero in nine. These comparisons describe the tested templates: the verbal endpoints are not assumed semantically equivalent to 100% and 0%, and sentence structure and objective order also differ.

The matched Breakthrough and Rush Hour comparisons use 150 positions per style across three depth strata, for 450 responses per style and priority level, pooled across models. In all six game–style pairs, each formulation yields a higher style-match rate at the strongest than at the weakest style priority.

The numerical formulation has the largest observed endpoint span in four of the six game–style pairs. In Breakthrough, the adverbial formulation has the largest span for right flank, and the modal formulation has the largest span for rear advance. Eight of the twelve pooled numerical-minus-verbal paired bootstrap intervals exclude zero in favor of the numerical formulation; the four comparisons for right flank and rear advance include zero. For rear advance, the numerical-minus-modal span difference is -0.044 [-0.096, 0.007], and the numerical-minus-adverbial difference is -0.018 [-0.071, 0.038].

We additionally compare the five condition means within each of the nine game–model trajectories in Figure 2. A condition mean is Pareto dominated when another priority level in the same trajectory has both an equal or higher style-match rate and an equal or higher strength-retention rate, with at least one strict inequality. None of the 45 numerical means is dominated. The modal formulation has 9 dominated means across five trajectories, and the adverbial formulation has 11 across six trajectories. These counts describe the observed point estimates; they do not constitute a statistical test of Pareto dominance.

Figures 10–13 disaggregate the single-decision responses by model, style, and instruction formulation. They separate the two outcomes so that each panel contains only the three formulation trajectories. Figure 14 gives the corresponding depth-resolved comparison. All formulations use the same

positions within each game, and lines connect observed conditions rather than fitted trends. The instructed endpoint shift recurs across the disaggregated verbal conditions, while their intermediate shapes vary more often than the numerical trajectories.

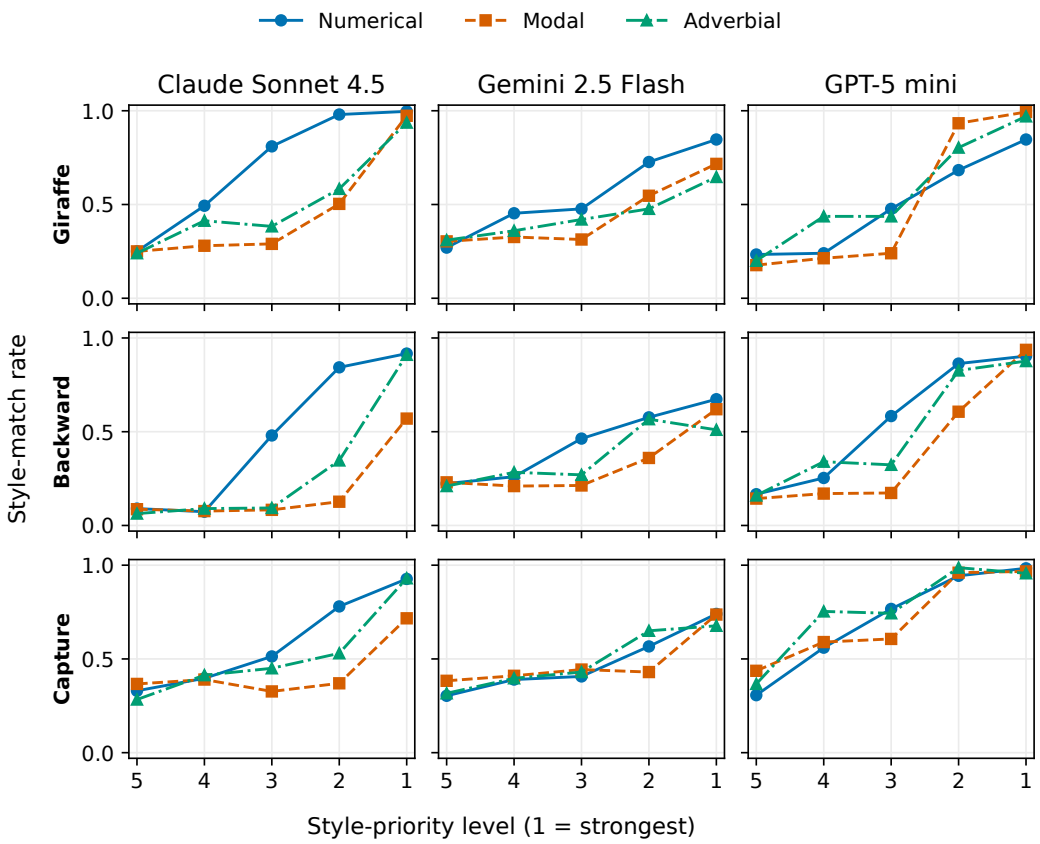

Figure 10: Animal Shogi style-match rates by model, style, and instruction formulation. Columns show models and rows show styles. Each point pools the three forced-win depths (300 decisions). The horizontal axis orders the five formulation-specific conditions from lowest (5) to highest (1) stated style priority.

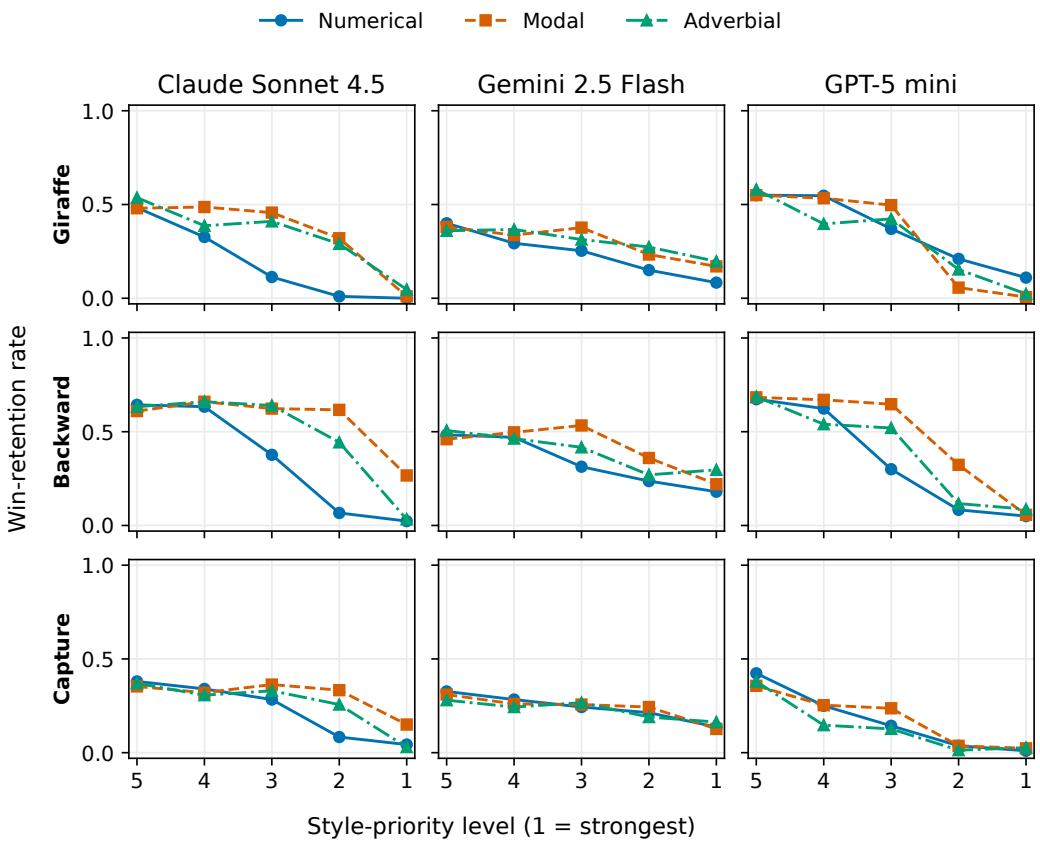

Figure 11: Animal Shogi win-retention rates by model, style, and instruction formulation, on the same positions and conditions as Figure 10. Each point pools the three forced-win depths (300 decisions).

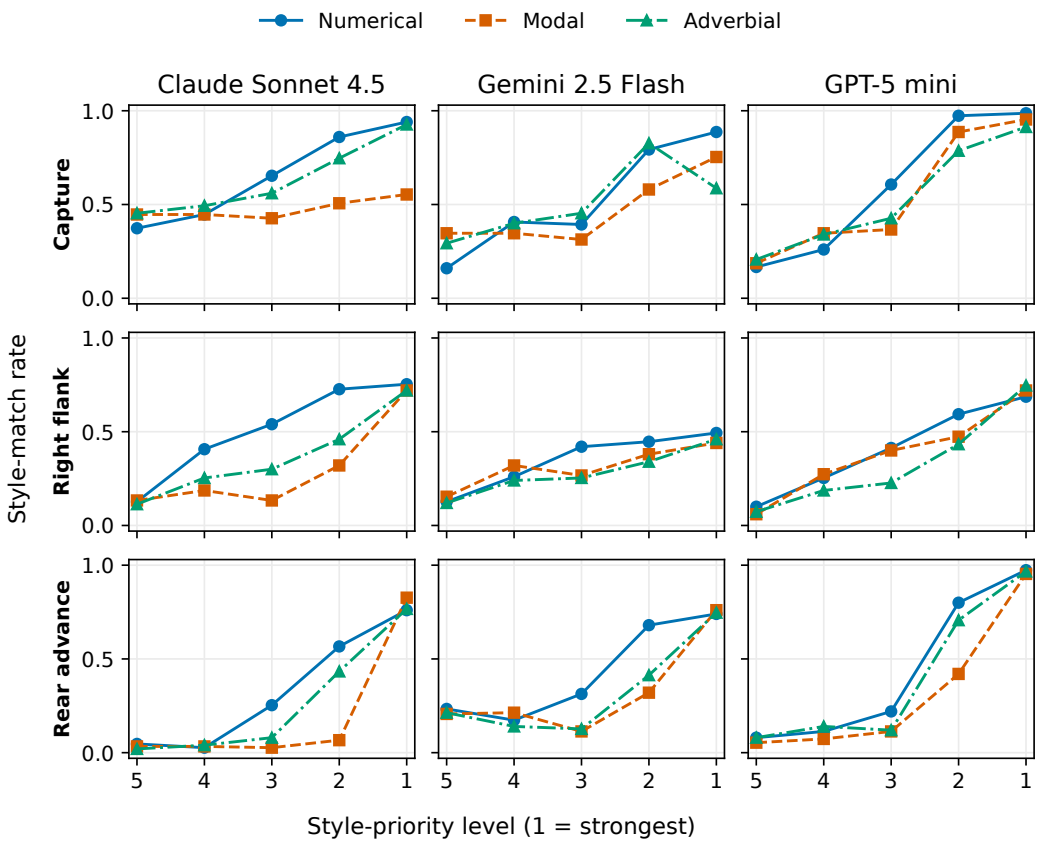

Figure 12: Breakthrough style-match rates by model, style, and instruction formulation. Columns show models and rows show styles. Each point pools the three forced-win depths (150 decisions). The horizontal axis orders the five formulation-specific conditions from lowest (5) to highest (1) stated style priority.

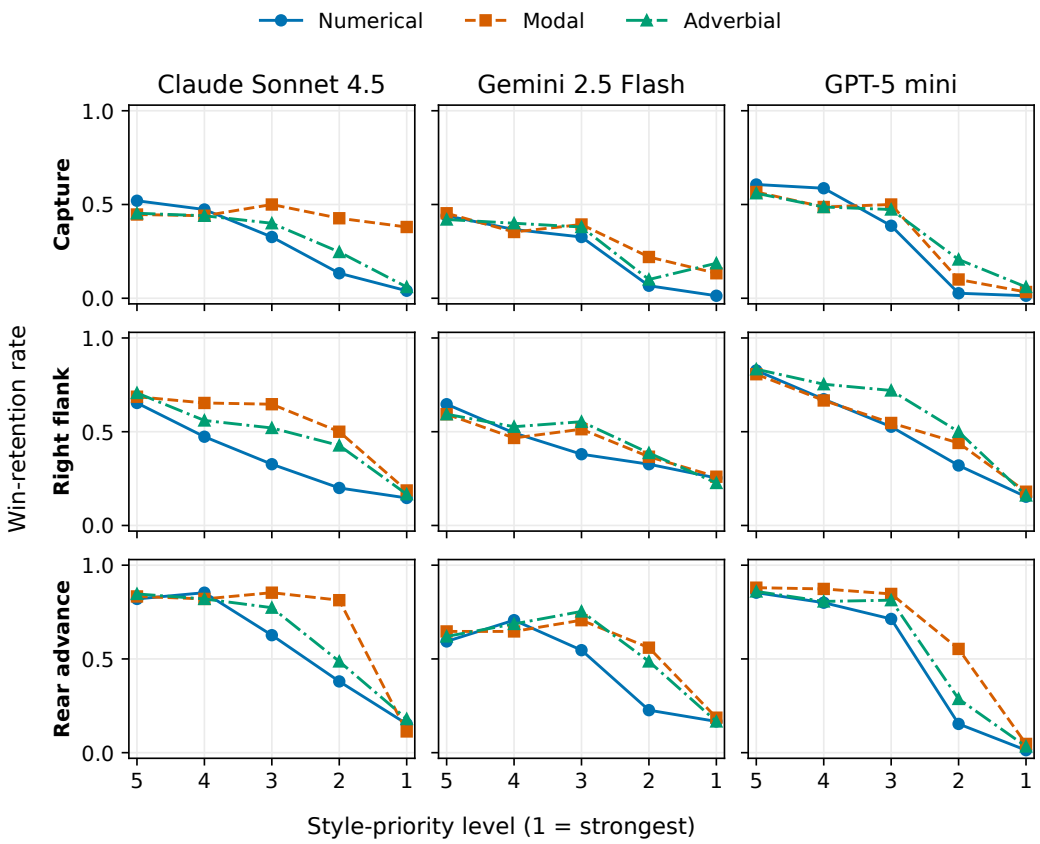

Figure 13: Breakthrough win-retention rates by model, style, and instruction formulation, on the same positions and conditions as Figure 12. Each point pools the three forced-win depths (150 decisions).

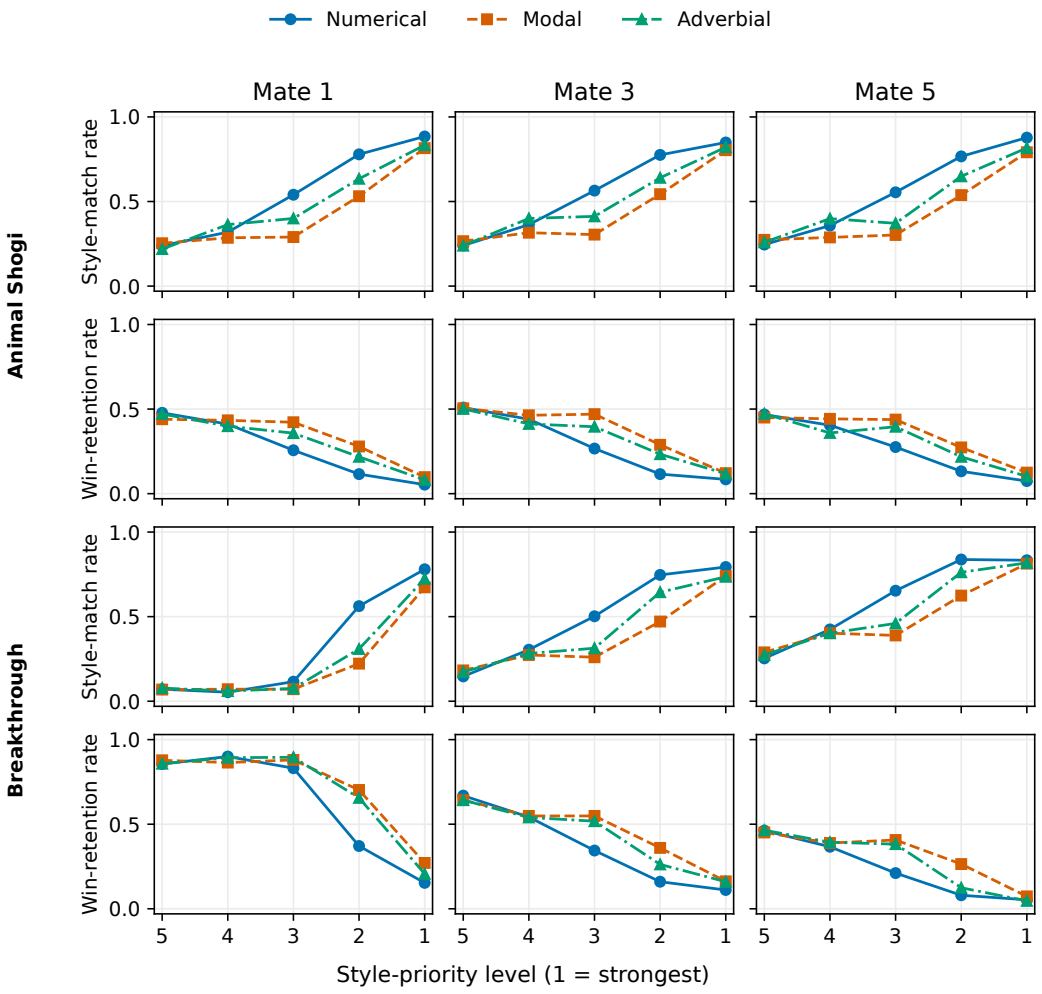

Figure 14: Style-match and win-retention rates by forced-win depth and instruction formulation. Columns show depth; paired rows show style-match and win-retention rates for each game. Each Animal Shogi point pools three models and three styles (900 decisions); each Breakthrough point pools three models and three styles (450 decisions). Lines connect observed conditions. Depth strata contain different positions, so their comparison does not isolate an effect of difficulty.

D.2 Varying style instructions with the winning instruction fixed

A separate Animal Shogi comparison tests whether changing only the style instruction shifts behavior while the winning instruction remains at its strongest tested level. We use the same three models, three styles, and 100 positions per style and forced-win depth (1/3/5). For each formulation, the fixed-win series varies style strength through five levels while holding winning strength fixed; the complementary series changes both strengths in opposite directions. The series share their weakest-style endpoint. The numerical fixed-win instruction assigns 100% to winning and varies the stated style percentage over 100/75/50/25/0; these two numbers need not sum to 100. The modal winning sentence is "You must play to win." and the adverbial sentence is "You very strongly prefer winning." The weakest style instruction explicitly assigns no priority to the style, as in the complementary comparison. At the strongest style level, the fixed-win series retains the strongest winning instruction, while the complementary series assigns no priority to winning. Numerical complementary responses reuse the main experiment.

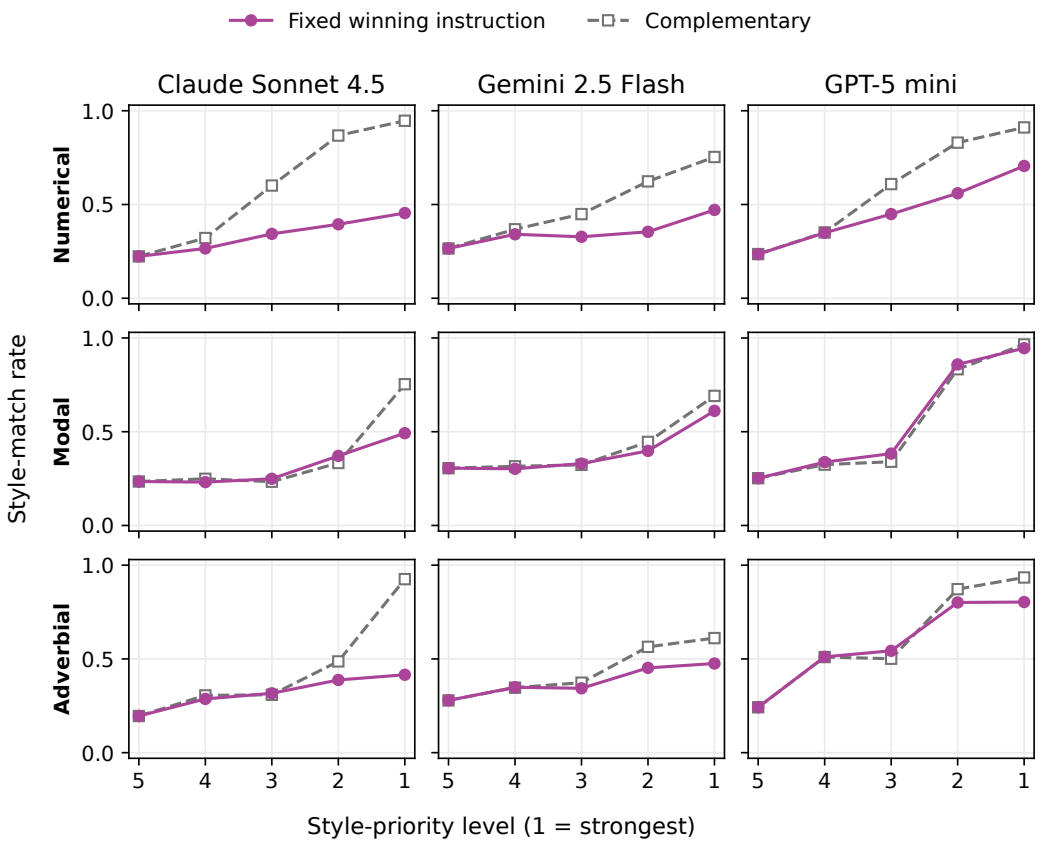

Figure 15: Animal Shogi style-match rates with the winning instruction fixed (purple, solid) or varied complementarily (gray, dashed). Rows show formulations; columns show models. Each point pools 900 decisions across three styles and three depths. Levels run from weakest (5) to strongest (1) style instruction, corresponding to 0/25/50/75/100% for numerical wording. The two series share the weakest-style responses, whose verbal instructions assign no priority to the style. At the strongest style level, the complementary verbal instructions assign no priority to winning, while the fixed-win instructions retain the strongest winning wording. Lines connect observed means.

Style matching responds even when the winning instruction is unchanged. Pooling the 900 position–style observations per model and level, the decrease in style-match rate from the highest to the lowest style priority is positive in all nine model–formulation combinations, ranging from 0.197 to 0.693; all nine pointwise 95% bootstrap intervals exclude zero. Win retention increases between these endpoints in all nine combinations, by 0.086–0.484. Intermediate style-match rates are not universally monotone, so these endpoint effects do not imply an ordered response at every adjacent level.

The complementary series has a larger style-match span than the fixed-win series in all nine model–formulation combinations (Figure 15), with all nine pointwise difference intervals above zero. The differences range from 0.206 to 0.492 for numerical wording, 0.020 to 0.261 for modal wording, and 0.131 to 0.510 for adverbial wording. Thus, changes to the style instruction alone elicit a substantial response, and changing the winning instruction as well increases the observed endpoint span across all tested formulations and models.

State-key joins verify full coverage, unique condition–state keys, literal system messages, and legality of all 72,900 recorded actions; recomputed rates reproduce all 729 source aggregates. Pointwise intervals use 5,000 paired board-bootstrap replicates (seed 20260926), stratified by depth and retaining all style observations of each sampled board with their original weighting. With one generation

per condition and position, these intervals describe board-sampling variation; they do not adjust for multiple comparisons.

D.3 SENTENCE-REPETITION CONTROL

This diagnostic tests whether mechanical duplication of the two objective sentences reproduces the graded response to explicit complementary priorities. It uses 50 positions for each of the Giraffe, Backward, and Capture goals, stratified across forced-win depths 1, 3, and 5, for 150 positions in total. The diagnostic uses the game sentence `You play to win.` and a style sentence of the form `Play style: {style goal}.` It tests game/style repetition-count pairs $(1,0),(3,1),(1,1),(3,3),(1,3),(0,1)$. For the four conditions containing both objectives, half of the selected positions receive the game sentences first and half receive the style sentences first. The assignment is fixed across models. The game-only condition is identical to the style-free baseline. All six conditions use the same positions, giving 2,700 decisions across the three models. The five-level percentage conditions are evaluated on these identical positions to provide the matched reference curve reported below.

The full style sentences below come from the condition specification, not reconstructed response logs. Repeat the game and style sentences the specified number of times, joining them with single newlines. Game-first places all game sentences before all style sentences; style-first reverses these blocks. No numerical sentence is added. The user-message format is unchanged.

```
Play style: Strongly prefer moves that involve your Giraffe (G). Use the Giraffe's moves or
↪  drops whenever legal, even at positional cost.

Play style: Strongly prefer moves that pull your pieces back toward your own home row (row 3).
↪  Avoid advancing when a retreat or lateral move is legal.

Play style: Prioritize capturing opponent pieces. Take any capture available over
↪  non-capturing moves, even if the capture looks positionally weaker.
```

Figure 16 reports the condition rates, and Table 6 reports the paired repetition contrasts.

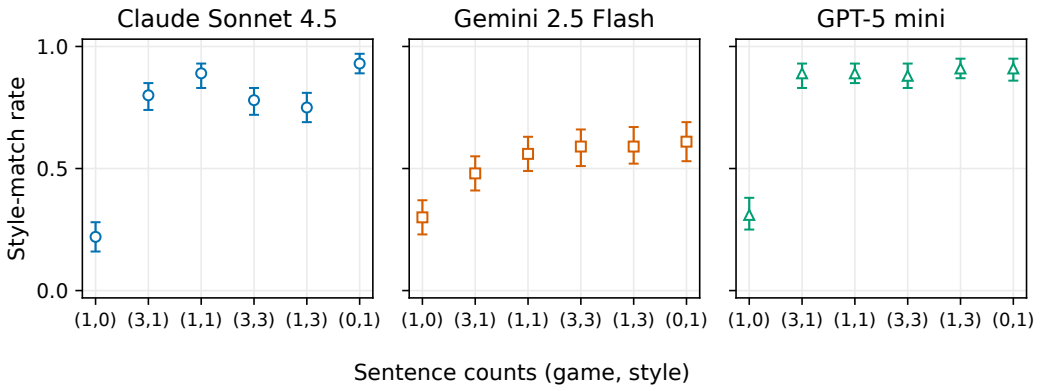

Figure 16: Style-match rates under the six sentence-repetition conditions. Points pool three styles and three depth strata; bars are pointwise 95% position-cluster bootstrap intervals from 2,000 replicates. Conditions are categorical, and no fitted trend is implied.

The count-derived style share does not order the observed rates. In particular, Claude selects style-matching moves less often in $(1,3)$ than in $(1,1)$, and $(1,1)$ and $(3,3)$ differ despite having the same ratio.

In contrast, the explicit percentage conditions on these positions yield wide, ordered five-level responses: Claude rates .93, .86, .59, .31, .21, Gemini rates .83, .58, .48, .34, .29, and GPT-5 mini rates .91, .83, .59, .34, .17 from full style to full game priority. GPT-5 mini's rates under the four both-objective repetition conditions remain high (.88–.91). Thus, mechanically repeating the objective sentences does not reproduce the graded response to the percentage instruction on the same positions.

Table 6: Paired changes in style-match rate after adding two copies of an objective sentence. Intervals use 2,000 position-cluster bootstrap replicates over 150 Animal Shogi positions. Counts are (game sentences, style sentences).

Comparison	Claude	Gemini	GPT-5 mini
$(1,3)-(1,1)$: add style sentences	-0.14 $[-0.20, -0.07]$	0.03 $[-0.05, 0.11]$	0.02 $[-0.02, 0.07]$
$(3,1)-(1,1)$: add game sentences	-0.09 $[-0.15, -0.03]$	-0.08 $[-0.18, 0.01]$	-0.01 $[-0.06, 0.05]$
$(3,3)-(1,1)$: preserve equal counts	-0.11 $[-0.18, -0.03]$	0.03 $[-0.07, 0.13]$	-0.01 $[-0.06, 0.03]$

Sentence order was counterbalanced across positions rather than crossed within every position, so we treat the order split descriptively. Claude's game-first and style-first rates are respectively .91/.69 for $(3,1)$, .92/.85 for $(1,1)$, .84/.72 for $(3,3)$, and .88/.61 for $(1,3)$. The corresponding Gemini rates are .49/.47, .57/.55, .57/.60, and .68/.51; GPT-5 mini's are .92/.85, .89/.89, .92/.84, and .93/.89. These differences show that arrangement may matter, but the diagnostic was not designed to identify a position-level causal order effect.

E Rush Hour episode analysis

All three models use the same 15 boards (IDs m01–m15 in the released board manifest), with three repeats at each of 15 style–priority conditions and one shared baseline: 2,160 episodes. Initial exact distances are 6–10 and move caps are $\max(20, 3d_0)$. Each move uses a fresh request with the same condition instruction and no response history. For each model and condition, we average the three repeats within board, then form pointwise 95% intervals from 5,000 paired bootstrap samples of the 15 boards, retaining all repeats together. Style wording was fixed using a disjoint five-board pilot.

E.1 Style–completion trade-offs

Across the five ordered conditions, all nine model–style trajectories show an overall graded shift toward greater mean episode style fractions, lower solve rates, and more wasted slides (Figure 17). Every paired endpoint interval excludes zero in these directions.

The intermediate trajectories are nonuniform. For GPT-5 mini's target-car style, the 50% to 75% shift raises style fraction by 0.370 [0.281, 0.459] and lowers solve rate by 0.844 [0.644, 1.000]. The solve fraction declines on 13 of the 15 boards, is unchanged on two, and improves on none, so this aggregate drop is not driven by only a few boards. With three repeats per board, these counts are descriptive rather than tests of board-specific effects.

E.2 Repetition and lack of solution progress

We replay all actions and recompute the exact number of one-cell slides needed to solve the puzzle at every visited state. An immediate reversal returns to the state before the preceding action; its denominator is the number of actions after the first. A revisit returns to any previously visited state, including the initial state. Style+other is the joint fraction of actions that match style without strictly decreasing the remaining solution distance. Fractions are computed within episodes, then averaged equally over repeats and boards. A non-decreasing move is not necessarily pointless, so these measures are descriptive rather than a causal decomposition of failure.

Full style priority increases both reversal and revisit rates relative to full solution priority in all nine model–style pairs. For GPT-5 mini's target-car 50% to 75% transition, the style+other fraction rises from 0.128 to 0.451. Thus much of the added style frequency comes from actions that do not reduce solution distance.

For this transition, paired board-bootstrap differences are 0.664 [0.505, 0.809] for immediate reversals, 0.641 [0.491, 0.777] for revisits, and 0.322 [0.244, 0.394] for style+other. These are descriptive associations, not a causal decomposition of failure. All 2,160 episode keys are unique and

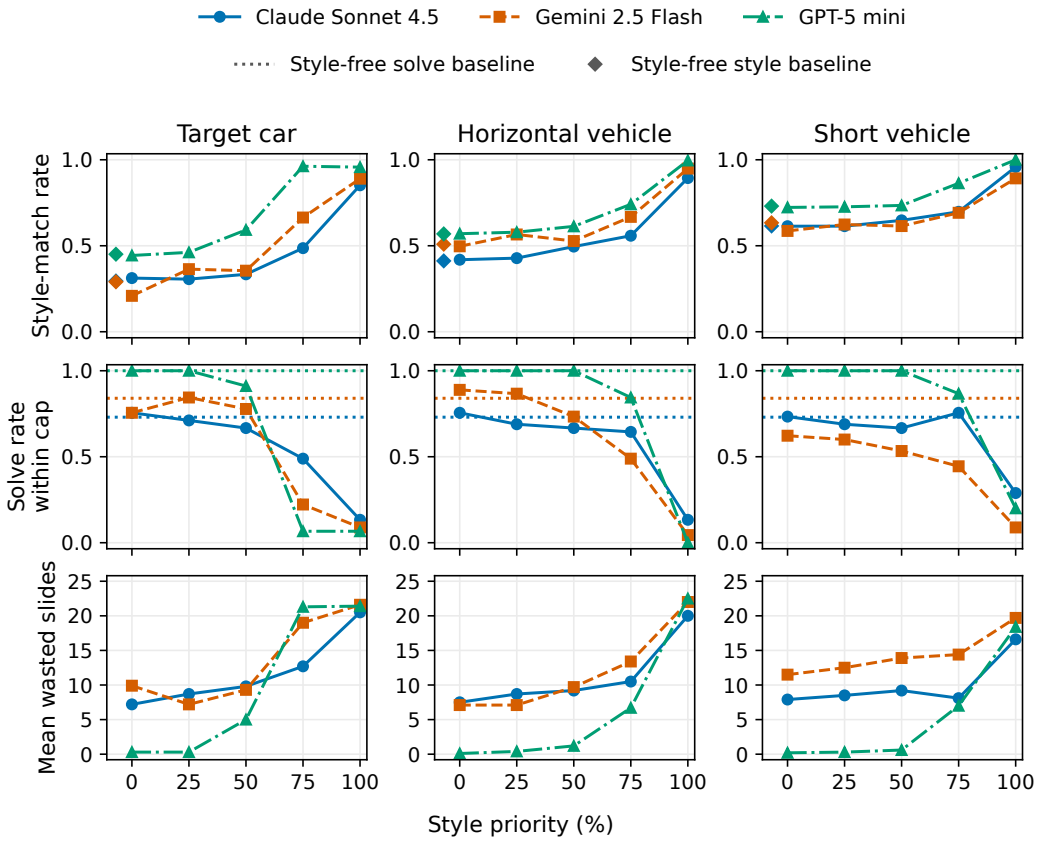

Figure 17: Rush Hour responses at all five priority levels. Columns show styles; colors, markers, and line patterns identify models as in Figure 2. Rows show style-match rate (the mean fraction per episode), solve rate, and mean wasted slides. Filled diamonds just left of 0% show style-free style baselines; dotted lines in the middle row show style-free solve baselines. Each point averages three repeats on 15 boards. Lines connect observed conditions and are not fitted trends.

the condition grid is complete. Replaying all 33,473 selected moves confirms legality, termination, and success flags; six unparsed moves use the prespecified first-legal-action fallback. An independent shortest-path implementation reproduces the reported style, solve, and wasted-slide aggregates within their source precision.

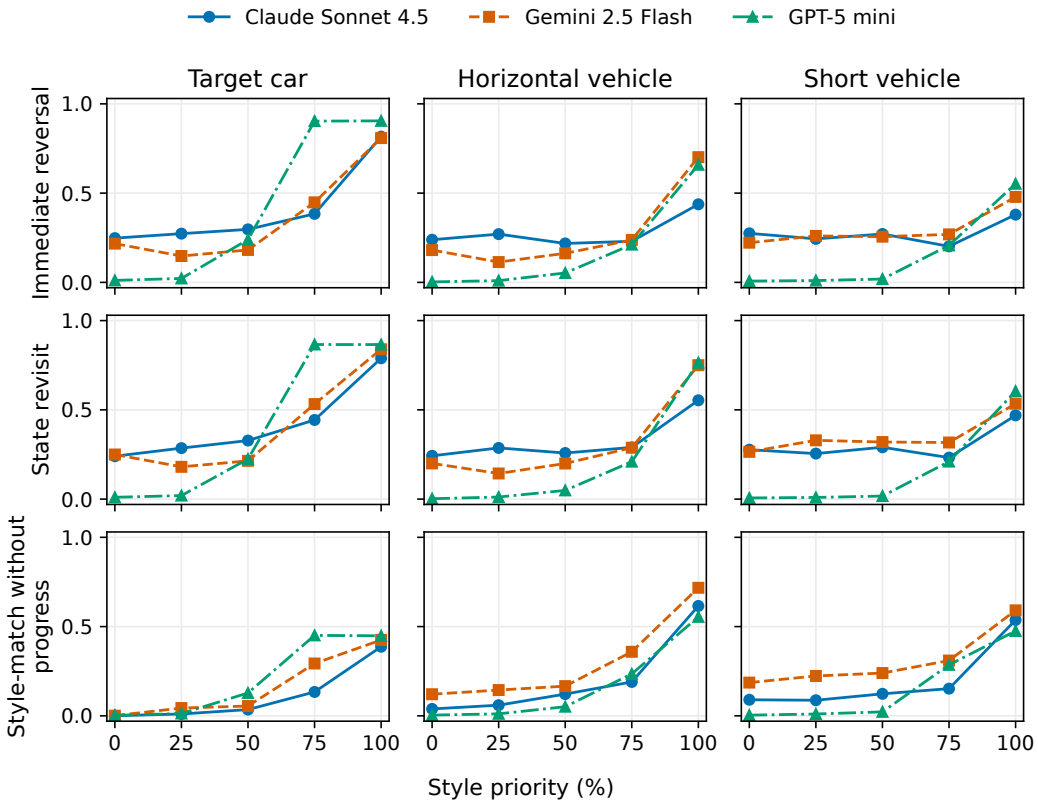

Figure 18: Rush Hour immediate reversals, state revisits, and style-matching actions that do not reduce remaining solution distance. Columns show styles; colors, markers, and line patterns identify models as in Figure 2. Fractions are averaged within repeat and board before aggregation; lines connect observed conditions and are not fitted trends.

F Repeated inference at individual positions

F.1 Repeated-inference protocol and aggregate response

To determine how the aggregate response in Figure 2 appears in position-level response distributions, we draw one model-independent set of 25 Giraffe positions from the shared candidate pool and evaluate the same board states for all three models. The draw is allocated across forced-win depths (nine positions at depth 1 and eight each at depths 3 and 5) and is otherwise unstratified. Sharing positions permits board-by-board model comparisons.

We repeat each priority condition 100 times per selected position. This estimates a response probability at each of the five priority levels and yields 37,500 responses: three models, 25 positions, five levels, and 100 repetitions. Every expected condition is present, all recorded actions are legal, and no batch error or duplicate condition label occurs. The plotted outcome is a selected action that matches the Giraffe style and incurs an exact value loss of at least 0.5, changing a theoretical win into either a draw or a loss. Every legal Giraffe-style action on these 25 conflict positions meets this loss criterion, so the plotted rate equals the style-match rate reported in Figure 3.

When averaged over the same 25 positions, the estimated event rate follows all five priority levels in order for each model (Figure 3). From 100% to 0% style priority, the means are 0.992, 0.968, 0.804, 0.474, and 0.241 for Claude; 0.881, 0.702, 0.530, 0.398, and 0.262 for Gemini; and 0.888, 0.690, 0.449, 0.220, and 0.210 for GPT-5 mini. These selected-subset summaries do not replace the main aggregate estimates, which use the full 300-position Giraffe sample per model and one generation

per position–priority condition. This follow-up instead isolates generation variability and estimated response shape on a common set of board states.

All 75 position–model pairs produce both this event and its complement under at least one of the five priorities. Moreover, its estimated rate is higher at 100% than at 0% style priority for all 75 pairs. Hence the aggregate response cannot be explained only by a deterministic partition in which some positions always elicit style and others always elicit strength. Within every tested position, higher style priority increases the estimated style-match rate between the endpoints. This is a statement about repeated-generation distributions, not a claim about an identified internal randomization mechanism.

F.2 Competition among concrete actions

We also ask whether this redistribution is mostly a contest between one exact-optimal action and one style-matching value-loss action. For each position–model pair, we pool its 100 generations at each endpoint and rank actions by their combined frequency. We call a pair single-action if its top action accounts for at least 90% of the 200 generations, two-action competition if its top two actions account for at least 90% and consist of one exact-optimal and one style-matching value-loss action, and other otherwise.

Table 7: Action competition on the 25 shared board states. Classification pools the two endpoint conditions (200 generations per position–model pair) and uses a 90% concentration threshold. Top-action change counts pairs whose most frequently selected concrete action differs between the 100% and 0% style-priority conditions.

Model	Single action	Style–optimal pair	Other	Top-action change
Claude	0/25	2/25	23/25	22/25
Gemini	0/25	0/25	25/25	20/25
GPT-5 mini	0/25	2/25	23/25	22/25
All	0/75	4/75	71/75	64/75

Only 4 of 75 position–model pairs concentrate at least 90% of their pooled endpoint responses on one exact-optimal action and one style-matching action (Table 7). No pair concentrates at least 90% on a single action. The most frequently selected action changes between endpoints in 64 of 75 pairs.

F.3 Position-level response shapes

Against the ordered aggregate response, Figure 3 shows substantial position-level heterogeneity in style matching. The point estimates form a non-increasing five-level profile for 24 of 25 Claude positions, 19 of 25 Gemini positions, and 16 of 25 GPT-5 mini positions. Across the 100 adjacent comparisons per model, respectively 1, 8, and 9 style-match point estimates decrease as style priority increases. The largest binomial standard error for any position–level estimate is 0.05 because each estimate uses 100 generations.

Win-retention is non-decreasing as style priority decreases for 20/25 Claude, 13/25 Gemini, and 10/25 GPT-5 mini positions. Together, these diagnostics show that a graded aggregate response does not imply smooth or ordered point estimates at every position.

Exact optimality is stricter than preserving a theoretical win because it excludes slower winning actions. As a supplementary check, its mean rate rises from the 100% to 0% style-priority endpoint for all three models: from 0.001 to 0.292 for Claude, 0.020 to 0.241 for Gemini, and 0.050 to 0.343 for GPT-5 mini. Complete non-decreasing profiles occur for 19/25, 12/25, and 8/25 positions, respectively. We use win retention for the competence condition below.

F.4 Conditioning on repeated win retention

We next ask whether the response remains after restricting attention to positions on which the model repeatedly preserves a theoretical win at 0% style priority. To avoid using the same generations to define and evaluate the result, we use all 100 generations at 0% style priority only to select positions

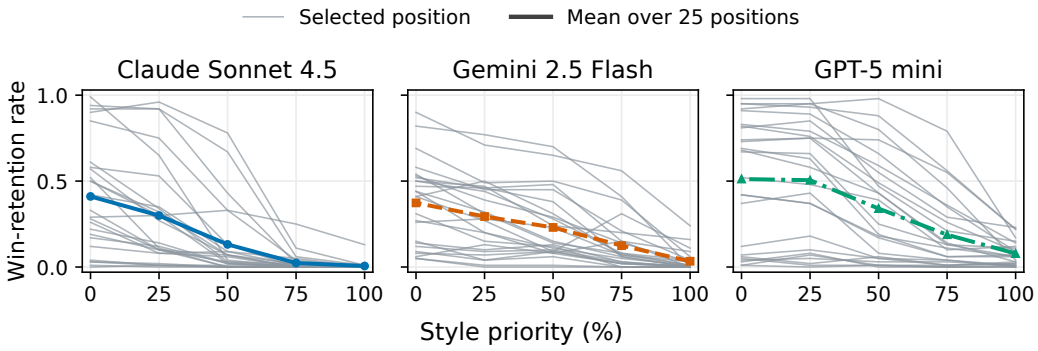

Figure 19: Position-level win-retention rates under 100 repetitions of the same request at each priority level on the same 25 Giraffe board states for all three models. Each thin line follows one selected position; the thick line is the equally weighted mean over the 25 positions for that model.

whose win-retention rate is at least 0.50. We then evaluate style-match rates using the independently generated 25%, 50%, 75%, and 100% conditions. The 0% point is omitted because it defines the subset.

The graded response on the fixed, repeatedly evaluated positions persists after competence conditioning. The criterion retains 11/25 positions for Claude, 9/25 for Gemini, and 14/25 for GPT-5 mini. On these subsets, both the style-match and win-retention rates remain ordered across the four independently evaluated levels for every model (Figure 20).

From 100% to 25% style priority, the style-match rates are 0.986, 0.950, 0.707, and 0.327 for Claude; 0.880, 0.609, 0.399, and 0.290 for Gemini; and 0.866, 0.654, 0.370, and 0.106 for GPT-5 mini. This remains a focused sensitivity analysis rather than a new population estimate.



Figure 20: Repeated Giraffe style-match (top) and win-retention (bottom) rates. Subsets retain positions with at least 50% win retention over 100 generations at 0% style priority. The independently evaluated 25–100% conditions are plotted; the 0% condition used to select the subset is omitted. Gray curves use all 25 positions.

F.5 CHOICE-SET AVAILABILITY

The residual Giraffe rate at 0% style priority is related to how many legal actions match the style (Figure 21). Across the 25 shared positions, the Pearson correlation is 0.77 for Claude, 0.70 for Gemini, and 0.53 for GPT-5 mini (Spearman correlations 0.68, 0.60, and 0.42). Positions offering more Giraffe actions therefore tend to retain more style-matching choices even when the instruction assigns full priority to winning.

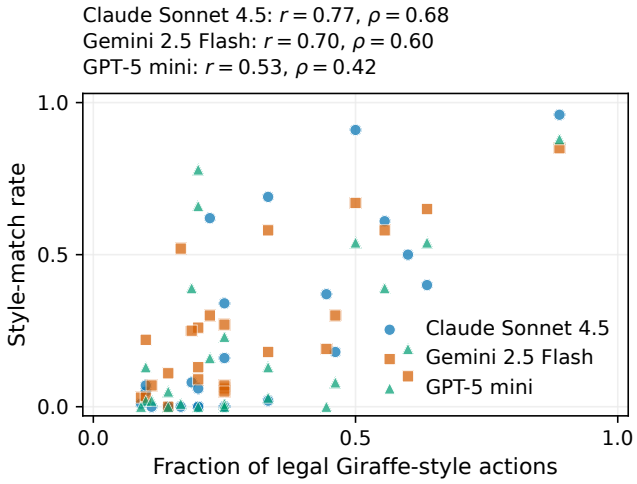

Figure 21: Residual Giraffe style matching versus action availability on the 25 shared positions. Each point is a position–model pair. The horizontal axis is the fraction of legal Giraffe moves or drops; the vertical axis is the style-match rate over 100 generations at 0% style priority.