The Contingency Bottleneck: The Emergence of Protocol Competence in Small Language Models for Interactive Language Games

Anonymous ACL submission

Abstract

Interactive learning assumes that models can already engage in meaningful, adaptive exchanges with other agents. Such exchanges require **contingency**: the ability to recognise and exploit structured dependencies between one's own actions and another agent's responses. This relies on a more basic **protocol competence**, a foundational capability underlying interactive learning: before an agent can learn from interaction, it must first learn how to participate in the interaction itself by following interaction rules, produce valid actions, and responding appropriately. Our experiments investigate the protocol competence of models from the Qwen model family, establishing a possible developmental sequence for interactive post-training, where demonstrations scaffold game participation, allowing models to learn from experience and refine credit assignment. Across five clembench dialogue games using Qwen3.5-0.8B, we find that 51.3% of base-model interactions fail before reasoning can be assessed due to protocol violations. This bottleneck is not solved by scaling or instruction tuning alone. We find *scaffolded teacher demonstrations bootstrap protocol competence for SLMs*, with substantial increases in interactive performance (+204% improvement on clem-score). Additionally, *teacher demonstrations "unlock" benefits of self-improving reinforcement learning* for SLMs in the clembench environment, with a single round of self-generated reinforcement learning (self-ReST) producing large gains in interactive performance from a relatively small number of successful trajectories. Among adaptive learning strategies for SLMs, credit-weighted training yields the most reliable improvements by reducing variance compared with uniform supervision.

1 Introduction

For interaction to serve as a learning signal, an agent must first establish **contingency** (Salhan et al., 2025): the ability to recognise and exploit structured dependencies between its own actions and another agent's responses. Contingency is what makes feedback *informative* rather than merely present: a reply, a correction, or a reward only teaches if the learner can relate it to the action that produced it and let it condition what it does next. Existing approaches to interactive post-training largely presuppose this capacity, treating interaction as a ready-made learning signal. However, in Classical AI, an action schema specifies not only *effects* but also *preconditions* that determine when the action is meaningful at all, distinguishing planning within an environment from the conditions that make action possible. We similarly contend that contingent interaction has preconditions that need to be established. We describe one prerequisite as **protocol competence**. In interactive games, before an agent can detect what its actions cause, it must produce actions the environment can respond to: entering the interaction in the required format, producing valid moves, maintaining the exchange structure across turns, and treating feedback as something that should alter future behaviour. A model that violates the protocol never generates the action–response pairs from which contingency could be learned; the interaction loop closes before it opens. Protocol competence is therefore not the same as solving the task: a model can be interaction-ready yet unable to win, or possess the relevant knowledge yet fail to participate long enough to access the learning signal.

With experiments on SLMs from the Qwen model family (Yang et al., 2025) across five clembench dialogue games (Chalamalasetti et al., 2023), we empirically investigate when SLMs acquire the competence to enter, maintain, and interpret an interaction protocol — the precondition for contingency — before they can exploit the feedback that protocol generates. We find a developmental sequence for interactive post-training in SLMs in which models must first learn to participate before

they can effectively learn from interaction. Our experiments establish that protocol competence is a bottleneck for SLMs, with 51.3% of base model episodes for our student model terminating due to protocol violations before reasoning can be evaluated. Across 160 evaluation settings, no model with near-zero participation can meaningfully achieve success, and is it not something that is consistently resolved by scaling (<7B parameter scales) or by instruction tuning. Interactive capabilities of models are highly non-monotonic from 0.8B to 7B parameters, and instruction-tuned models continue to fail on many protocol-constrained games.

We find that teacher demonstrations "unlock" these capabilities in models. Demonstrations from a teacher model about how to participate in an interactive game increase CLEMScore from 6.67 to 14.97 through improved adherence to interaction protocols. Once reliable participation is established, interaction becomes a useful learning signal: a single round of self-ReST further increases performance to 20.27 CLEMScore (+204% over the base model), while a static capability probe changes only marginally, suggesting that the gains are specific to interactive behaviour rather than general reasoning ability. Finally, after protocol competence has been acquired, further improvements come from more effective use of interaction experience, with credit-weighted learning producing the most stable adaptation.

2 Background

Interactive language learning. Most LLM post-training optimises behaviour from *static* supervision, either by imitating demonstrations (SFT) or learning from fixed preference data (DPO). Playpen (Horst et al., 2025) instead studies whether language models can improve through *interaction*. Built on clembench (Chalamalasetti et al., 2023), it formulates dialogue games such as Wordle, Taboo, and reference games as self-play environments in which a Game Master enforces the protocol and the game outcome provides an automatic reward signal. The `clemscore` performance is a product of **playrate**, a measure of successful completion of the interaction protocol, and **quality**, which measures task performance conditioned on completed episodes. Playpen showed that reinforcement learning over self-play generalises better than supervised imitation for an instruction-tuned 8B model. However, these experiments assume a learner that already participates reliably in interaction: formal protocol violations are rare, so failures primarily reflect task reasoning rather than participation itself.

Scaffolding and interactive learning. Developmental theories distinguish between learning a task and learning to participate in the activity through which the task is performed. Vygotsky's Zone of Proximal Development (ZPD) describes tasks that become solvable with appropriate support (Vygotsky, 1978), while scaffolding provides temporary structure that enables participation before independent competence emerges (Wood et al., 1976). Applied to interactive language learning, these ideas suggest that demonstrations should primarily establish participation in the interaction protocol, after which self-generated experience can support further improvement. We test this hypothesis through protocol-focused teacher demonstrations and adaptive self-improvement strategies inspired by the ZPD.

3 Methodology

Our central hypothesis is that successful interactive learning requires a prerequisite we call *protocol competence*: the ability to maintain the dialogue protocol, produce valid actions, interpret environmental feedback, and remain engaged in the interaction. Without this competence, a model cannot obtain meaningful learning signals regardless of its underlying reasoning ability. Once protocol competence is established, subsequent improvement should depend primarily on ordinary experience-driven learning rather than continued teacher supervision.

3.1 Interactive Games

We evaluate on five dialogue games from clembench, chosen to span different forms of interactive reasoning while enforcing strict communication protocols. Taboo requires constrained paraphrasing under alternating `CLUE:/GUESS:` turns; Wordle requires sequential constraint reasoning over feedback from previous guesses; ImageGame involves collaborative reconstruction of a spatial grid through incremental natural-language instructions; ReferenceGame tests discriminative referring expressions over visually similar alternatives; and PrivateShared evaluates maintenance of common ground during multi-turn dialogue. In every game, protocol violations immediately terminate

the episode, allowing participation failures to be distinguished from task failures.

3.2 Experimental Setup

Unless otherwise stated, the learner is Qwen3.5-0.8B (Yang et al., 2025), a sub-billion-parameter base model selected to study the emergence of interaction independently of instruction tuning or RLHF. Teacher demonstrations are generated by Qwen2.5-3B-Instruct, with alternative teachers evaluated as controls. We report clemscore together with its two components: play-rate as a measure of protocol competence and quality as a measure of task reasoning. Every checkpoint is additionally evaluated on a static non-interactive benchmark to distinguish improvements specific to interaction from general capability gains.

3.3 Conditions for Interactive Learning

We investigate three questions.

First, can teacher demonstrations establish protocol competence? We fine-tune the learner on demonstrations that illustrate valid interaction behaviour, including response formatting, turn-taking, and feedback handling, while measuring improvements in participation separately from task success.

Second, once a model can participate, can it continue learning without a teacher? Following ReST (Gulcehre et al., 2023a), the learner generates self-play trajectories (Grow), retains successful episodes (Select), and fine-tunes on these trajectories (Improve), yielding a fully offline self-improvement procedure.

Third, how much successful experience is required for autonomous improvement? We vary the number of successful trajectories used for training ({64, 128, 256, 512}) to measure the sample efficiency of interactive self-learning.

3.4 Conditions for Adaptive Learning

Having established protocol competence, we investigate how self-generated experience should be used. We compare standard ReST, which weights all successful episodes equally, against three adaptive alternatives under matched compute. First, *ZPD-based selection* preferentially trains on successful episodes that are challenging but attainable for the current learner. Second, *credit-weighted learning* assigns greater weight to trajectories expected to produce larger behavioural updates. Third, decoding controls vary sampling during trajectory generation to separate exploration effects from learning effects. Finally, we compare offline self-improvement against online policy optimisation with GRPO and GiGPO (Shao et al., 2024; Feng et al., 2025), which perform credit assignment during interaction rather than after data collection.

3.5 Alternative Explanations

To distinguish protocol competence from competing explanations, we evaluate several counterfactuals. We vary model scale (0.5B–7B) to test whether interaction emerges purely from additional capacity; compare base and instruction-tuned checkpoints to isolate the effect of instruction tuning; vary the quantity of training data to separate supervision volume from supervision type; compare full, protocol-only, and reasoning-only demonstrations to distinguish structural guidance from solution transfer; vary teacher identity to assess teacher dependence; and sweep decoding parameters to control for exploration. Finally, we relate initial play-rate to subsequent learning gains to test whether protocol competence acts as a threshold for interactive learning.

3.6 Transfer Beyond Training Games

Finally, we ask whether protocol competence generalises beyond the games on which it is acquired. Models trained on ReferenceGame, Taboo, and ImageGame are evaluated on held-out dialogue games to determine whether interactive learning transfers to previously unseen protocols rather than merely improving performance on familiar tasks.

4 Experiments & Results

Our experiments reveal a simple but important constraint on interactive learning: *models cannot learn from interaction until they first learn how to participate in it*. Focusing on the Qwen family, we find base SLMs fail predominantly at *entering* the interaction protocol rather than at reasoning within it (§4.1), a deficit that neither scale nor instruction tuning resolves in small scales (§4.2). However, we find a small budget of teacher demonstrations scaffolds protocol competence in SLMs (§4.3), and once participation is established, self-generated experience becomes an effective training signal (§4.4), credit assignment stabilises adaptation within the established channel rather than creating it (§4.5).

4.1 Protocol Competence is a bottleneck for SLMs

Interactive environments impose a new requirement beyond language modelling: models must produce valid actions, maintain the interaction state, and follow the protocol expected by the environment. Without this ability, feedback never becomes available. We first examine how base models fail before any interactive learning takes place. Across 2,210 episodes, more than half terminate before reasoning can even be evaluated. As shown in Table 3, protocol violations account for 1,133 failures (51.3%), exceeding reasoning failures among episodes that did participate successfully (763 episodes, 34.5%). The dominant errors are not incorrect solutions, but failures to enter the game correctly: producing extra text (33.8%), invalid actions (10.7%), invalid values (6.3%), or malformed responses. Only 14.2% of episodes end in success. The failures of base SLMs in interactive games suggests that there is a divergence between *knowing how to solve a task* and *knowing how to participate in the process that solves the task*.

Table 1: Failure decomposition across 2,210 evaluation episodes. Most failures occur before reasoning is evaluated because models fail to participate in the interaction protocol.

Outcome	Count	%
Protocol failures		
Extra text	746	33.8
Invalid action	237	10.7
Invalid value	138	6.3
Missing tag	12	0.5
Protocol subtotal	**1133**	**51.3**
Reasoning failure	763	34.5
Success	314	14.2
Total	**2210**	**100.0**

Decoupling Participation from Performance. Teacher demonstrations primarily address failures of model participation rather than task competence itself. Table 2 reveals a consistent dissociation between learning *how to enter and maintain a game protocol* (play-rate) and learning *how to succeed within that protocol* (win-rate). The effect of demonstrations depends strongly on the initial capability profile of each task. For games where participation is already near ceiling—Reference, Image, and Taboo, with base play-rates of 1.00, 1.00, and 0.97 respectively—there is little protocol deficit to repair, and demonstrations instead improve downstream performance: win-rates increase from 33.3 to 61.1 for Reference, from 0.0 to 8.2 for Image, and from 0.0 to 5.8 for Taboo. In contrast, where protocol competence is missing, repair can be necessary but is not sufficient for success. Image provides the clearest example: play-rate increases from 0.06 to 1.00 after demonstrations, while win-rate remains unchanged at 0.00. Reference is the only task where repairing participation translates into substantial performance gains, with play-rate reaching 1.00 and win-rate increasing from 0.40 to 0.61. At the other extreme, PrivShared remains unaffected: neither participation nor performance improves, suggesting that demonstrations do not universally establish the required interaction protocol.

The Wordle Dissociation. Wordle provides the strongest evidence that protocol acquisition and task competence are separable capabilities. Task-specific demonstrations were required to elicit valid Wordle gameplay: general-purpose examples left participation at 0/30 episodes, whereas 400 Wordle-specific examples raised it to 29/30. However, this gain reflected the model learning the strict output format rather than the ability to solve the game, as participation rose without producing a single win. A dedicated Wordle repair set containing 400 demonstrations increases play-rate from 0.00 to 0.97 (29/30 instances successfully entered), yet produces no wins. The model learns the interaction protocol required to participate by formatting valid moves, maintaining the exchange structure, and producing acceptable responses, but does not acquire the strategic competence required to solve the game. Crucially, this repair is highly targeted: the general-purpose demonstration pool used throughout our other experiments fails to improve Wordle participation, leaving play-rate at 0.00 in the standard mechanism comparisons (Table 8). Thus, protocol competence is not an automatic consequence of exposure to demonstrations; it requires supervision aligned with the specific interaction structure of the task. Together, these results suggest that demonstrations do not primarily teach solutions, but instead establish the interaction channel through which later feedback and learning can become effective.

Table 2: **Effect of Teacher Demonstrations in Dialogue Games:** We evaluated both the base and repaired models across all instances, meaning the columns can be directly compared. For games the base model already plays, its participation rate is maxed out before any additional training; in these cases, providing demonstrations mostly just improves the quality of its interactions. When the model fails to participate, it only improves if given training data specific to that game's rules. For example, 400 Wordle-specific demonstrations increased participation from 0/30 to 29/30 episodes (though without producing any wins), whereas the general-purpose demonstrations used for other tasks left Wordle participation at zero. None of the tested interventions managed to fix participation in the PrivateShared game.Quality is measured on a scale from 0 to 100.

	Play-rate		Quality	
Task	Base	Repaired	Base	Repaired
Reference	1.00	1.00	33.3	61.1
Image	1.00	1.00	0.0	8.2
Taboo	0.97	0.95	0.0	5.8
PrivShared	0.00	0.00	0.0	0.0
Wordle *(general)*	0.00	0.00	0.0	0.0
Wordle *(dedicated)*	0.00	**0.97**	0.0	**0.0**

Table 3: Failure decomposition across 2,210 evaluation episodes. Most failures occur before reasoning is evaluated because models fail to participate in the interaction protocol.

Outcome	Count	%
Protocol failures		
Extra text	746	33.8
Invalid action	237	10.7
Invalid value	138	6.3
Missing tag	12	0.5
Protocol subtotal	**1133**	**51.3**
Reasoning failure	763	34.5
Success	314	14.2
Total	**2210**	**100.0**

4.2 Impact of Model Size and Instruction Tuning

A natural hypothesis is that larger models already contain the ability to interact, and that failures simply reflect insufficient capacity. However, scale alone does not appear sufficient. Across five off-the-shelf models spanning 0.8B–7B parameters (Table 4; two Qwen3.5 models (Yang et al., 2025) and three Qwen2.5 models (Yang et al., 2024)), we observe that interactive ability is non-monotonic with model size. Three of the five model sizes collapse to a clemscore of zero, despite being larger than the smallest model. The smallest checkpoint achieves the highest deterministic play-rate (0.59) and a clemscore (6.67) within half a point of the best-performing checkpoint (`Qwen2.5-3B`, 7.04), while models nearly an order of magnitude larger fail to produce meaningful interactions.

Table 4: **Interactive ability does not reliably emerge from model scale alone.** Evaluation uses deterministic decoding to isolate model-level protocol competence from sampling-induced format instability (Appendix section C). Three of five Qwen base models have clemscores that collapse to zero, and the smallest model is the most reliable participant.

Model	Size	Play-rate	Fmt-abort	clemscore
Qwen3.5-0.8B	0.8B	0.59	0.60	6.67
Qwen2.5-1.5B	1.5B	0.00	1.00	0.00
Qwen2.5-3B	3B	0.38	0.47	7.04
Qwen3.5-4B	4B	0.00	–	0.00
Qwen2.5-7B	7B	0.17	0.81	0.00

4.3 Teacher demonstrations bootstrap interactive learning

Once the model can participate, interaction becomes informative. Table 5 summarises the effect of teacher demonstration, which **increases interactive performance from 6.67 to 14.97 clemscore, and subsequent self-ReST training raises it further to 20.27**. Overall, teacher demonstrations provide a +13.60 (+204%) improvement over base, with an overall increase interactive performance from 6.67 to 14.97 clemscore, and subsequent self-ReST training raises it further to 20.27 — a +13.60 (+204%) gain over base. Demonstrations supply +8.30 of the gain, and self-ReST the remaining +5.30 (+35% over the demonstration stage).

We use static probes to distinguish acquired protocol competence from general capability improvements. Unlike interactive evaluations, which provide multi-turn feedback and measure adaptation within an environment, these probes present fixed inputs with no external feedback and therefore measure one-shot knowledge and reasoning. The observation that interactive performance improves while static probe scores remain unchanged suggests that training primarily enhances the model's ability to engage with feedback-driven protocols rather than increasing its underlying task knowledge. The static probe moves only modestly over the same training (30.28 → 35.21 → 33.23, a net +2.95 / +10%) and does not track the interactive

trajectory: it peaks after demonstrations and then declines slightly under self-ReST even as interactive score continues to rise. Teacher demonstrations therefore do not primarily add new knowledge or general reasoning ability; they unlock the ability to learn from interaction. This confirms that gains from teacher demonstrations are concentrated in interaction-specific contexts.

Table 5: Teacher demonstrations establish the interaction channel, after which self-ReST enables further improvement. Gains are concentrated in interactive evaluation, not the static probe.

Training stage	Interactive score	Static probe
Base model	6.67	30.28
Teacher demonstrations	14.97	35.21
Self-ReST	20.27	33.23
Gain over base	+13.60 (+204%)	+2.95 (+10%)

4.4 After participation, models learn from experience

Once SLMs gain protocol competence from teacher demonstrations, models can now generate useful interaction trajectories and improve from its own successes and failures. Following ReST (Gulcehre et al., 2023b), we perform self-ReST as an offline Grow–Select–Improve procedure rather than an interleaved reinforcement learning loop. In each round, the current policy first generates candidate game trajectories (Grow), retains only successful interaction episodes (Select), and is then fine-tuned on these winning demonstrations (Improve). Generation, filtering, and optimization are therefore separated into distinct phases: the model does not update during rollout collection, and learning occurs only after the offline winner set has been constructed. Implementational details are in subsection B.1. Figure 1 shows that most gains occur during the earliest rounds of self-ReST, followed by diminishing returns. The quantity of self-generated supervision is distinct from the number of ReST rounds. In particular, our winner-budget analysis (Figure 1) varies the number of retained winning trajectories rather than tracking offline training rounds. Performance increases as additional winners are added up to a peak at $N = 256$ examples, after which larger winner pools introduce diminishing returns. Separately, the online ReST experiment shown in Figure 1b evaluates an on-policy optimisation loop and does not represent the offline Grow–Select-Improve procedure used for the main self-ReST result. This suggests that the largest barrier was not extracting better information from experience, but first becoming capable of producing informative experiences at all.

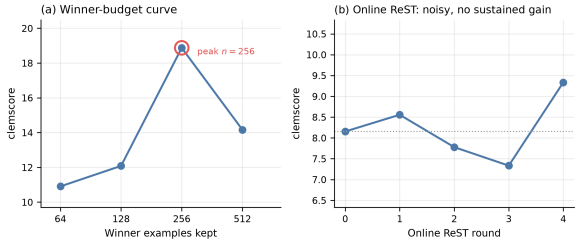

Figure 1: **Self-ReST learning curves:.** Improvement occurs rapidly once protocol competence has been acquired. The table reports the resulting CLEMScores after training on different numbers of winning trajectories. Each value corresponds to a single run.

Winning (N)	64	128	256	512
CLEMSCore	10.91	12.08	18.87	14.16

4.5 Once interaction exists, credit assignment affects stability

After models can participate, the remaining challenge is deciding which parts of an interaction trajectory were responsible for success.

All four methods in Table 6 implement the same offline self-improvement procedure: *Grow* by sampling k attempts per game instance from the current policy, *Select* successful trajectories, and *Improve* by fine-tuning on the retained winners. They differ only in how winners are selected and weighted during training. Let $\mathcal{W}$ be the set of valid winning trajectories and let the improvement objective be

$$\mathcal{L}(\theta) = -\sum_{\tau \in \mathcal{S}} \sum_{t} w(\tau, t) \log \pi_\theta(a_t | s_t),$$

where $\mathcal{S}$ is the retained training set and $w(\tau, t)$ controls the contribution of each assistant action. **Flat ReST** uses all valid winners with equal weighting, i.e. $\mathcal{S} = \mathcal{W}$ and $w(\tau, t) = 1$, corresponding to standard rejection-sampling fine-tuning. **ZPD selection** keeps the same training objective but filters winners to instances near the model's current ability level. For an instance with empirical win rate W_i, only trajectories from instances satisfying $\ell \leq W_i \leq h$ are retained, with $(\ell, h) = (0.25, 0.75)$. Thus, very easy instances (almost always solved) and very difficult instances (almost always failed) are removed because they provide less learning signal. **Credit-weighted ReST (offline GiGPO**

(**Feng et al., 2025**)) keeps the full winner set but replaces uniform weighting with a per-turn credit score:

$$w(\tau_i, t) = A_i^{\text{epi}} + A_{i,t}^{\text{step}}.$$

The episode-level term gives higher weight to wins that are difficult relative to other sampled attempts, while the step-level term rewards turns that make progress compared with alternative trajectories from similar states. This transfers GiGPO-style credit assignment from online policy optimisation into offline supervised fine-tuning. We then experiment with **decoding** strategies that keep the Flat ReST selection and weighting scheme unchanged, but vary the sampling temperature and nucleus probability during generation as a control condition, testing whether gains come from better exploration rather than from improved data selection or credit assignment. Since all methods use identical compute, winner filtering, and fine-tuning budgets, differences isolate the effect of the underlying mechanism. Comparing adaptation strategies under matched compute, Table 6) shows that credit-weighted learning attains the highest mean clemscore (19.17) and, most notably, the lowest variance (± 0.73 vs. ± 1.38–± 2.68 for the alternatives) — roughly a $3\times$ reduction in standard deviation relative to rejection sampling.

Table 6: Matched-compute comparison of adaptation mechanisms. Credit-weighting yields the highest mean and by far the lowest variance, though no mean difference is significant after Holm correction.

Method	clemscore	Δ vs. Flat	Holm p
ReST	18.30 ± 2.65	—	—
ZPD	17.93 ± 2.68	-0.37	1.00
Credit	$\mathbf{19.17 \pm 0.73}$	$+0.78$	1.00
Decoding	17.40 ± 1.38	-1.30	1.00

4.6 Analysis: What gates interactive learning

The results above suggest a developmental sequence for interactive learning: models must first acquire protocol competence before interaction itself becomes a useful learning signal. Appendix C analyses this claim in detail.

First, participation is a necessary precondition for success. Across all evaluation settings, no model with near-zero play-rate achieves meaningful performance, while once participation is established, quality almost perfectly predicts success (Table 8). The underlying failures are overwhelmingly interface failures rather than reasoning failures: models most commonly produce valid-looking responses that violate the expected protocol through extra text or malformed actions (Table 18), and these failures are highly decoding-sensitive (Table 7). Together, these results indicate that the principal bottleneck for small language models is entering the interaction, not reasoning within it.

Second, this bottleneck is not explained by either model scale or instruction tuning. Interactive ability is highly non-monotonic across the Qwen family (Table 4), while instruction-tuned checkpoints still fail to participate reliably on many games (Table 11). Extending evaluation across the wider clembench suite reveals that most games remain format-bound for the untrained models (Table 14), suggesting that protocol competence is an independent capability rather than a by-product of larger models or instruction following.

Teacher demonstrations primarily repair this interaction bottleneck by teaching protocol rather than solutions. Their effect is largely independent of the teacher used (Table 9), and protocol-only supervision substantially outperforms reasoning-only supervision (Table 10). Wordle provides the clearest dissociation: dedicated demonstrations almost completely restore participation without producing any successful games (Table 12), showing that learning to participate and learning to solve are separable abilities.

Once participation has been established, further improvements come almost entirely from better reasoning within the interaction. Stage-wise analyses show that subsequent self-ReST gains are concentrated in interaction quality rather than play-rate (Table 19), while static reasoning probes (Table 20), theory-of-mind evaluation (Table 13), and winner-budget experiments (Table 23) provide little evidence that training substantially changes underlying general reasoning ability. Instead, interaction training appears to improve how models exploit feedback after they can already participate.

Finally, credit assignment refines an existing interaction channel rather than creating one. Per-game decompositions attribute nearly all gains from credit weighting to reasoning improvements in games where participation is already solved (Table 8), while matched-compute comparisons show only modest average improvements over standard rejection sampling despite substantially lower variance (Tables 21–22). Similarly, online credit

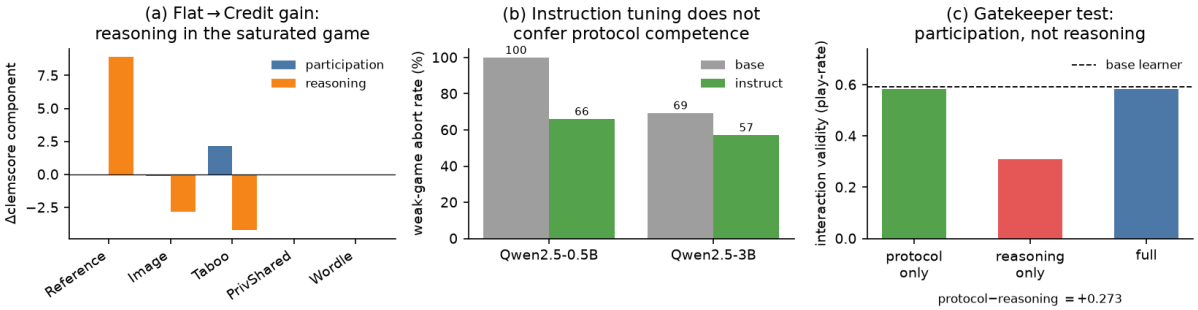

Figure 2: Experiments separating *protocol competence* from *reasoning ability*. **(a)** Shapley decomposition of ReSt →credit-weighted ReST CLEMSCORE gains shows that reasoning improvements emerge primarily in ReferenceGame, where participation is already high. **(b)** Instruction tuning reduces but does not eliminate protocol failures, indicating that it does not guarantee interactive competence. **(c)** Protocol-only supervision restores valid interactions, whereas reasoning-only supervision does not, showing that participation is a prerequisite for learning from feedback.

assignment fails to outperform the offline Grow–Select–Improve procedure and may even reduce protocol competence (Tables 24–25). The resulting capability also transfers only weakly. Models do not acquire a general interaction protocol that immediately applies to unseen games (Table 15), and improvements on novel instances remain mixed across games (Table 16). Rather than producing a general reasoning upgrade, interactive learning appears primarily to improve the model's ability to acquire and exploit task-specific interaction protocols.

5 Discussion and Conclusion

Interactive learning is not a capability that emerges from scale or instruction tuning, but a developmental process with distinct prerequisites. Models must first become capable of sustaining an interaction before they can exploit the information that interaction provides. Protocol competence is the mechanism that determines whether feedback can enter the learning loop at all. The prevalence of protocol failures (51.3% of base-model episodes) and the absence of success among settings with negligible participation suggest that many apparent failures of interactive reasoning are, in fact, failures to access the interactive regime. This perspective also changes how we interpret the role of demonstrations. Rather than simply transferring task solutions, demonstrations often function as a scaffold that establishes the missing interaction channel. By making the structure of acceptable behaviour explicit, they allow contingencies to become observable and exploitable. However, the uneven repair across games indicates that protocol competence is not a single transferable ability: some interaction patterns can be acquired from general scaffolding, while others require game-specific experience or remain beyond the learner's current competence frontier. Once this channel is established, experience becomes valuable. Self-ReST improves interactive performance without significantly altering other model capabilities or behaviours, suggesting that the primary gain comes from learning how to participate rather than acquiring new world knowledge. At the same time, our results reveal a tension in on-policy adaptation: the same training process that exploits interaction can also disrupt it. Maintaining a stable interaction channel may therefore be a central challenge for future interactive post-training methods. Credit assignment provides a complementary lesson. More sophisticated weighting mechanisms improve learning only after a model can already participate; they refine an existing learning signal rather than create one. The strongest gains appear in settings where the interaction loop is already reliable, suggesting that optimisation methods alone cannot overcome missing behavioural prerequisites. More broadly, our findings highlight evaluating interactive models not only by final task success, but by their ability to enter, maintain, and learn from structured exchanges. Interactive post-training may be better understood as a process of making models *teachable*: establishing the conditions under which feedback can shape future behaviour. Future work should therefore investigate methods that preserve participation during adaptation, design curricula for unrepaired interaction protocols, and measure how efficiently models acquire new forms of social and task-specific coordination.

6 Limitations

Single Model Family All learners and teachers are Qwen (Qwen3.5, Qwen2.5). The claim that protocol competence is a distinct, scale-independent capability rests on five checkpoints from one family; other families (Llama, Gemma, Phi, OLMo) may show different participation profiles, especially given how idiosyncratic format-following is to pretraining data mixes.

Single learner for all training experiments. Every intervention (repair, self-ReST, credit assignment, transfer) is run only on Qwen3.5-0.8B. The scale sweep is evaluation-only; you never test whether the *training* findings (demonstrations→ReST staging, credit-weighting variance reduction) hold at 3B or 7B.

Single benchmark family with strict abort semantics. All games are clembench, where any protocol violation terminates the episode. These are text-only, synthetic, English-only games with scripted or model interlocutors. Our description of protocol competence may partly be an artifact of clembench's unusually strict parser; environments with lenient parsing, reprompting, retries, or soft grading (or real users, who repair misunderstandings) may not show the same gate.

Experimental Confounds GRPO/GiGPO get four rounds, one hyperparameter setting, and (for the single-turn run) one game whose participation is already saturated.

References

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for LLM agent training. *arXiv preprint arXiv:2505.10978*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023a. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023b. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. Playpen: An environment for exploring learning through conversational interaction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Suchir Salhan, Hongyi Gu, Donya Rooein, Diana Galvan-Sosa, Gabrielle Gaudeau, Andrew Caines, Zheng Yuan, and Paula Buttery. 2025. Teacher demonstrations in a BabyLM's zone of proximal development for contingent multi-turn interaction. In *Proceedings of the First BabyLM Workshop*, pages 323–355, Suzhou, China. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Lev S. Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA.

David Wood, Jerome S. Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

A Description of Games

Taboo is a word-description game played between two model instances. Player A (the describer) is given a target word together with a list of semantically related "taboo" words it must not use; Player B (the guesser) must identify the target. Turns alternate under a strict format: the describer's turns must begin with CLUE: and the guesser's with GUESS:, and the game master parses each turn against these templates. The game runs for at most three guess rounds and is aborted if either player violates the format or the describer utters the target or a forbidden word. Success is scored by speed (100/t for a correct guess at turn t; 0 otherwise). The game probes lexical knowledge and paraphrastic description under negative constraints, plus the ability to adapt clues after a failed guess.

Wordle is a letter-based word-guessing game in which a single model player must identify a hidden five-letter word within six attempts. Feedback comes from a deterministic game-master bot, which annotates each guessed letter as green (correct position), yellow (in the word, wrong position), or red (absent), rendered textually. Each turn the model must emit a guess in a rigid two-field format (a guess: line containing exactly one valid five-letter word, plus an explanation), and malformed responses lead to reprompting and ultimately to an aborted episode. The benchmark includes two assisted variants — Wordle+clue, where a semantic hint accompanies the puzzle, and Wordle+clue+critic, where a second model player critiques the proposed guess before it is submitted — but the core demand is the same: integrating positional letter feedback across turns while never breaking the output format. Scoring again uses the speed metric.

ImageGame (the drawing-instruction game) is a two-player collaborative construction task. Player A, the instruction giver, sees a target 5×5 character grid and must convey it to Player B, the instruction follower, who starts from an empty grid and updates it incrementally in response to A's natural-language instructions (e.g., filling particular cells with particular letters). A signals completion with DONE, and the episode is capped at a maximum number of turns; each of B's turns must render the current grid state so the game master can track progress. Target grids come in two flavours — compact, patterned grids that reward abstract, efficient descriptions, and random grids that force cell-by-cell instruction. Performance is scored by the F1 overlap (precision/recall over filled cells) between the target grid and B's final grid, so the game measures spatial-language generation on one side and faithful instruction following and state maintenance on the other.

ReferenceGame is a Lewis-style signalling (reference resolution) game. Both players are shown the same three 5×5 grids, presented in independently shuffled order; Player A is told which grid is the target and must produce a single referring expression for it, and Player B must then say which of its three grids the expression picks out. The exchange is a single turn per player, with templated outputs (an expression from A, a first/second/third answer from B) that the game master parses strictly. Scoring is binary — did B select the target grid — with no partial credit. The game tests whether A can distil a discriminative, concept-level description of a visual pattern (rather than an exhaustive one) and whether B can ground that description against competing distractors.

PrivateShared is a dialogue scorekeeping game with a single model player. The model acts as a customer holding a set of private slot values (for instance, the details of a travel booking) while a questioner — played by the game master from a script — elicits them one by one, moving information from the private to the shared ground of the conversation. Interleaved with these questions, the game master poses side probes off the main dialogue record, asking whether the interlocutor already knows a given piece of information; the model must answer yes or no, and all responses must carry the required tags (e.g., ANSWER: / aside markers), with format violations after reprompting aborting the episode. Scoring combines slot-filling accuracy in the main dialogue with agreement on the probes (chance-corrected via Cohen's), taken as a harmonic mean. The game thus tests whether a model maintains an explicit discourse model — tracking not just what is true, but what has been shared with whom, while simultaneously honouring a two-channel (in-dialogue vs. meta-level) response protocol.

B Implementational Details

B.1 Implementation of ReST

We sample trajectories from the current policy with temperature 0.8 and $k = 4$ attempts per instance over the five-game Playpen suite. The filtering stage retains only normalized wins containing a valid assistant interaction turn and at least two messages, removing degenerate successful traces. To avoid over-representing easier games, surviving trajectories are balanced across games, and the headline self-ReST results use flat weighting where each retained winner contributes equally to supervised fine-tuning. Importantly, self-ReST is not applied to the raw base model. The procedure begins from a protocol-repaired checkpoint obtained through teacher demonstrations: the model first improves from a base clemscore of 6.67 to 14.97 through supervised protocol repair, after which self-ReST further increases performance to 20.27 (Table 5). Thus, self-ReST operates after the model has acquired basic interaction competence: the Grow phase explores within an already functional protocol, rather than attempting to discover the interaction format from scratch. Each Improve stage performs full-parameter supervised fine-tuning for one epoch with learning rate 10^{-5} and batch size 16. The previous checkpoint is retained as a regression safeguard, and a new policy is promoted only when it improves the benchmark score. The resulting gain should therefore be interpreted as extracting additional capability from self-generated successful interactions, rather than repairing failures of participation.

C Additional Experiments and Analyses

Decoding effects on protocol participation. The base model's play-rate varies substantially depending on the evaluation procedure. This difference does not reflect inconsistent estimates of the model's underlying competence, but rather the sensitivity of interaction protocol adherence to decoding stochasticity. Under rollout sampling ($k = 6$, $T = 0.9$), the model frequently produces additional conversational text or otherwise violates the required response format, leading to protocol aborts. This effect is particularly pronounced in ImageGame, where play-rate decreases from 1.00 under deterministic decoding to 0.06 under sampled rollouts, and in Taboo. In contrast, greedy decoding strongly constrains generation to the expected format, revealing that the model can enter these protocols when format variance is removed. We therefore report deterministic full-instance evaluation for direct comparisons of protocol competence, while using sampled rollouts only where interaction diversity is required (Table 7).

Table 7: **The same base model measured two ways.** Left: sampled rollout (k=6, T=0.9), the distribution self-ReST actually trains on. Right: deterministic single-pass evaluation over the full instance set, the protocol used for every clemscore in this paper. The two disagree sharply on ImageGame and Taboo: under sampling the model frequently emits extra prose and aborts, while a single greedy pass usually stays in format. The macro-average play-rate is 0.33 under sampling and 0.59 deterministically — these are two real measurements of different quantities, not a discrepancy.

	Play-rate	
Game	Rollout (k=6, T=0.9)	Deterministic eval
ReferenceGame	0.96	1.00
ImageGame	0.06	1.00
Taboo	0.58	0.97
Wordle	0.04	0.00
PrivateShared	0.00	0.00
Macro-average	0.33	0.59

C.1 Participation is a prerequisite for successful interaction

Across 160 evaluation settings, we use play-rate as a measure of protocol competence. This measure captures whether a model can successfully enter the interaction protocol by producing valid actions and maintaining the required exchange structure. Play-rate is perfectly correlated with the inverse of format-abort rate ($r = 1.000$), confirming that it measures successful participation rather than task success.

The distinction between participation and reasoning is particularly clear at the extremes. Among settings with extremely low participation (play-rate < 0.05), no successful outcomes are observed. In contrast, once models participate reliably, reasoning quality becomes predictive of success ($r = 0.99$). Thus, interaction failures are not simply failures of problem solving: models must first acquire the ability to enter the interaction loop before reasoning improvements can matter. A Shapley decomposition of the Flat→Credit gap makes the same point from the other direction: the credit gain is a reasoning gain concentrated in the one game whose participation is already saturated (Table 8).

Table 8: Per-game decomposition of the Flat→Credit clemscore gap (paired, n=7 seeds). Play/Quality shown as flat/credit. Δ is the paired mean gap; † marks Holm-corrected $p < 0.05$ across the five games. The gap is Shapley-split into a *participation* ($\frac{1}{2}\Delta p\,(q_f{+}q_c)$) and a *reasoning* ($\frac{1}{2}\Delta q\,(p_f{+}p_c)$) component that sum to Δ exactly. The credit gain is a *reasoning* gain concentrated in the one game whose participation is already saturated (ReferenceGame, play≈1), i.e. it appears exactly where protocol competence is already established.

Game	Play f/c	Qual f/c	Flat	Credit	Δclem	Δparticip.	Δreason.
ReferenceGame	1.00/1.00	71.4/80.3	71.43	80.32	+8.89	+0.00	+8.89
ImageGame	0.99/0.98	14.0/11.1	13.86	10.91	-2.94	-0.09	-2.85
Taboo	0.67/0.91	11.2/5.6	7.14	5.12	-2.02	+2.18	-4.20
PrivateShared	0.00/0.00	0.0/0.0	0.00	0.00	+0.00	+0.00	+0.00
Wordle	0.00/0.00	0.0/0.0	0.00	0.00	+0.00	+0.00	+0.00
Overall (macro)			18.49	19.27	+0.78	+0.42	+0.37

D Teacher demonstrations and protocol learning

A central question is whether demonstrations improve interaction because they transfer a teacher's general instruction-following ability or because they expose the learner to the structure of successful interaction. We separate these explanations by varying teacher models while keeping the learner, demonstration budget, training procedure, and evaluation fixed. Protocol improvements remain robust across teachers, including teachers without instruction tuning (Table 9). This indicates that demonstrations are effective because they provide examples of valid interaction patterns rather than because they transfer the conversational behaviour or instruction-following ability of a stronger model.

Table 9: Protocol repair is teacher-robust and not merely inherited instruction-following. Base learner clemscore 6.667. Only the teacher varies (base vs instruct, weak→strong); the demo cap, learner, SFT and eval are fixed.

Teacher	Tuning	Scale	#demos	clemscore	Δvs base
Qwen2.5-3B	base	3B	50	6.67	+0.00
Qwen2.5-0.5B-Instruct	instruct	0.5B	88	6.22	-0.44
Qwen2.5-3B-Instruct	instruct	3B	136	9.11	+2.44

D.1 Protocol supervision versus reasoning supervision

To isolate what information demonstrations provide, we compare protocol-only demonstrations, reasoning-only demonstrations, and full demonstrations. Protocol-only demonstrations preserve valid interaction structure while removing solutions. Reasoning-only demonstrations provide solutions but do not explicitly teach the interaction protocol. If protocol competence is the main bottleneck, protocol supervision should improve participation more than reasoning supervision. This is what we observe: protocol-only supervision produces larger gains than reasoning-only supervision despite using the same demonstration budget (Table 10).

D.2 Instruction following is not equivalent to interaction competence

Another possibility is that protocol failures arise only because the evaluated models are not instruction tuned. To test this, we compare matched base and instruction-tuned checkpoints without additional interaction training. Instruction tuning improves general instruction following, but does not eliminate

Table 10: Protocol repair versus reasoning repair. All conditions use the same demonstration budget. The base learner begins with play-rate $= 0.593$ and clemscore $= 6.67$.

Condition	n	Play-rate	ΔPlay	Clemscore	ΔClem
Protocol-only	5	0.582	-0.011	8.57	+1.90
Reasoning-only	5	0.309	-0.284	7.32	+0.65
Full	5	0.583	-0.010	9.61	+2.94

interactive failures (Table 11). Even instruction-tuned models continue to produce invalid actions, violate required formats, or fail to maintain the interaction state. These results suggest that instruction following and interaction competence are related but distinct abilities. Instruction tuning teaches models to respond appropriately to instructions, whereas interactive competence additionally requires maintaining a structured exchange, producing valid actions, and using feedback from previous interaction steps.

Table 11: Protocol failure is not an instruction-tuning artifact. Raw checkpoints (no training) evaluated on the interaction suite. Even instruction-tuned models abort the protocol on most weak-game instances; instruction tuning helps but does not confer protocol competence, and the same aborting pattern holds for the paper's Qwen3.5-0.8B *base* learner.

Model	Tuning	Weak-game abort	Overall abort	clemscore
Qwen2.5-0.5B	base	100%	100%	0.00
Qwen2.5-0.5B-Instruct	instruct	66%	50%	6.33
Qwen2.5-3B	base	69%	47%	7.05
Qwen2.5-3B-Instruct	instruct	57%	38%	11.33

D.3 Participation enables learning from experience

The previous experiments show that demonstrations establish the ability to participate. We test whether interaction training produces broad improvements beyond interactive environments. Protocol repair can restore participation without improving task success. For example, in Wordle, repairing the interaction format increases participation from zero to almost complete participation but produces no wins. Similarly, theory-of-mind evaluation remains unchanged after interaction training. These results indicate that interaction training does not provide a general reasoning improvement. Instead, it teaches a more specific capability: learning effectively from feedback and experience within structured interactive environments. Across evaluation settings, play-rate is strongly associated with eventual success. Extremely low participation prevents meaningful learning, whereas once interaction is established, improvements arise from better reasoning. A per-task analysis demonstrates this separation: repairing the interaction protocol can restore valid participation without necessarily improving task solving (Table 19; see also Table 2 in the main text). This distinction is particularly clear in Wordle: format repair increases participation from zero to almost complete participation, but produces no successful games (Table 12). Thus, participation and problem solving represent distinct stages of interactive learning.

Table 12: Wordle participation frontier. Protocol repair enables valid interaction without improving task success.

Condition	Played	Wins	Clemscore
Before repair	0/30	0	0.00
After repair	29/30	0	0.00

E Credit assignment improves reasoning after participation exists

Once models can participate, self-generated experience becomes useful. Teacher demonstrations first repair participation, after which self-ReST can use successful trajectories as training data. This creates a progression from a model unable to reliably interact, to one capable of generating useful experiences, to one that can improve from those experiences. Credit assignment provides a further test of this hypothesis.

If credit assignment improves the ability to interact, gains should appear primarily in participation metrics. Instead, improvements are concentrated in settings where participation is already saturated, particularly ReferenceGame, where play-rate is already approximately one. This shows that credit assignment improves learning within an existing interaction channel rather than creating that channel. Once models can reliably participate, the remaining challenge is extracting useful learning signals from interaction trajectories. We compare alternative adaptation strategies under matched compute. Credit weighted learning achieves the highest average performance, but the improvement is not caused by increased participation. A game-level decomposition shows that the gain is concentrated in settings where protocol competence is already saturated. In ReferenceGame, where play-rate is already one, credit assignment improves solution quality substantially (Table 8). This indicates that better learning signals become useful only after the model has acquired the ability to interact.

E.1 Interaction learning is specialised

Finally, we test whether interaction training produces broad improvements in unrelated cognitive abilities. Static evaluation remains largely unchanged after interaction training (Table 20), and theory-of-mind false-belief accuracy is unaffected (Table 13). These results suggest that interaction training does not produce a general reasoning upgrade. Instead, the learned capability is more specific: models become better at acquiring useful behaviour from feedback and experience within interactive environments.

Table 13: Interaction training does not improve unrelated theory-of-mind evaluation.

Checkpoint	False-belief accuracy	N
Qwen-base	0.473	300
Format repair	0.473	300
Best (+ReST)	0.473	300

F Additional Games

We use additional CLEMBench games that are not part of the main training benchmark to additionally provide an external characterisation of whether the observed protocol failure phenomenon is specific to the five core games or reflects a broader property of interactive evaluation. These additional environments allow us to investigate when a small language model fails an interactive task, does it fail because it cannot reason about the underlying problem, or because it cannot successfully enter and maintain the interaction protocol required by the environment? The first analysis is a taxonomy study over a broader set of CLEMBench games (Table 14).

F.1 Out-of-Distribution Protocol Competence

We evaluate the untrained base policy across 13 additional games and classify failures according to whether the dominant bottleneck is protocol compliance or task reasoning. A game is considered format-bound when the model rarely produces a valid move, meaning that learning signals from the environment cannot be exploited because the interaction never properly begins. In contrast, a game is reasoning-bound when the model participates successfully but fails to solve the underlying task. The results show that protocol failures dominate: 10 of 13 games are format-bound, including Wordle variants, PrivateShared, Codenames, GuessWhat, and TextMapWorld variants. Only a small number of games, including MatchIt (ASCII), ReferenceGame, and Taboo, are primarily reasoning-bound. This supports the broader claim that interactive evaluation often measures a prerequisite capability—entering and sustaining the protocol—before it measures task-specific reasoning ability.

Table 14: **Which games are protocol-bound for a sub-1B base model.** Untrained base policy across the wider clembench suite. A game is *format-bound* if the model rarely produces a valid move at all, and *reasoning-bound* if it participates but loses. 10 of 13 games are format-bound: for most of the suite, a small base model never reaches the point where a reward signal exists.

Game	n	Play-rate	clemscore	Class
DoND	40	0.38	0.0	format-bound
Wordle (with critic)	30	0.13	0.0	format-bound
Codenames	130	0.00	0.0	format-bound
GuessWhat	60	0.00	0.0	format-bound
PrivateShared	50	0.00	0.0	format-bound
TextMapWorld	50	0.00	0.0	format-bound
TextMapWorld (graph)	30	0.00	0.0	format-bound
TextMapWorld (room)	30	0.00	0.0	format-bound
Wordle	30	0.00	0.0	format-bound
Wordle (with clue)	30	0.00	0.0	format-bound
MatchIt (ASCII)	40	1.00	40.0	reasoning-bound
ReferenceGame	63	1.00	33.3	reasoning-bound
Taboo	60	0.97	0.0	reasoning-bound

F.2 Does Protocol Competence Transfer between Games?

The second analysis studies whether protocol competence transfers between games (Table 15). We train the model on a subset of interactive games (ReferenceGame, Taboo, and ImageGame) and evaluate it on unseen games. The results indicate limited zero-shot transfer: held-out games such as Wordle and PrivateShared remain at zero play-rate because the model does not successfully initiate the interaction protocol in the new environment. Consequently, previously learned interaction strategies cannot be applied, since the model never reaches the stage where feedback, rewards, or task-specific reasoning become available. This finding supports the view that protocol competence is not simply a general reasoning ability that automatically transfers across interfaces; instead, successful adaptation requires acquiring the conventions of the new interaction format.

Table 15: **Transfer to held-out games.** Trained on ReferenceGame, Taboo, ImageGame (n=226 winners), then evaluated on games never seen in training. Training games improve; every held-out game remains at zero, because each held-out game is one the base model could not enter in the first place (cf. Table 14). We therefore do *not* claim protocol transfer to unseen games.

Game	n	Play-rate	Quality	clemscore
Training games				
ReferenceGame	90	1.00	86.7	86.7
Taboo	60	0.98	7.6	7.5
ImageGame	40	0.97	9.2	8.9
Held-out games (never trained on)				
Wordle	30	0.00	0.0	0.0
PrivateShared	50	0.00	0.0	0.0
GuessWhat	60	0.88	15.1	13.3
Codenames	130	0.00	0.0	0.0

F.3 Few-Shot Protocol Competence

A complementary few-shot adaptation experiment investigates whether an interaction-trained model can rapidly acquire a new protocol once given demonstrations. GuessWhat is used as a held-out adaptation setting: rather than testing pure transfer, the experiment measures how many demonstrations are required for the model to cross a basic participation threshold. The interaction-trained model adapts substantially faster than the base model, reaching competent participation with few or no additional demonstrations. This result suggests that once a model has acquired general interaction competence, learning a new protocol may become easier. However, this experiment is limited to a single game and a single threshold, so it should be interpreted as suggestive evidence for improved adaptability rather than a definitive claim about universal transfer. A related question is whether the training-game gains themselves reflect transferable skill or the training distribution: on novel instances, ReferenceGame quality collapses while Taboo quality rises, ruling out simple memorisation only for the latter (Table 16).

Table 16: **Are the wins skill or distribution?** The best model (`qwen-fmt-rest`) evaluated on the original instances vs. novel instances of the same game. ReferenceGame quality collapses on novel grids ($82.2 \rightarrow 35.6$), so a large part of that game's gain is specific to the training distribution. Taboo moves the other way (quality rises on novel words, on a smaller played subset), which rules out simple word memorisation there.

Game	Instances	Play-rate	Quality	Wins
ReferenceGame	original	1.00	82.2	74/90
	novel	1.00	35.6	32/90
Taboo	original	0.43	7.7	2/60
	novel	0.28	94.1	16/60
Wordle	original	0.00	0.0	0/30
	novel	0.00	0.0	0/30

G Additional Analysis: Diagnosing the Bottleneck in Interactive Learning

G.1 Base Models Fail Primarily at Entering the Interaction Protocol

Across 2,210 sampled base-model episodes, 51.3% terminate in a protocol abort before any verdict is reached, with PrivateShared and Wordle almost entirely protocol-bound and ReferenceGame almost entirely reasoning-bound (Table 17).

Table 17: **Where base-model episodes die.** Base policy (Qwen3.5-0.8B), k=6 sampled rollouts at T=0.9 over five games, n=2210 episodes. *Play-rate* = (win+loss)/total: the share of episodes that reached a verdict rather than aborting on a protocol violation. Two games (PrivateShared, Wordle) are almost entirely protocol-bound; ReferenceGame is almost entirely reasoning-bound.

Game	n	Abort (fmt)	Loss	Win	Play-rate	Win-rate|played
ReferenceGame	792	28	458	306	0.96	0.40
ImageGame	455	429	26	0	0.06	0.00
Taboo	483	203	272	8	0.58	0.03
Wordle	180	173	7	0	0.04	0.00
PrivateShared	300	300	0	0	0.00	0.00
Overall	**2210**	**1133**	**763**	**314**	**0.49**	**0.29**

G.2 Protocol Failures Are Mostly Interface Violations Rather Than Reasoning Errors

The single largest violation subtype is a valid move wrapped in extra commentary (189 of 287 adjudicated cases), and format and reasoning failures occur at the same median relative position in the episode, ruling out late-episode degradation as the cause (Table 18).

Table 18: **What protocol violations look like.** Counts of adjudicated format-violation subtypes. *Extra prose* — the model emits a valid move wrapped in commentary — is the single largest category, and dominates Wordle. Format and reasoning failures occur at the same median relative position in the episode (0.333 vs 0.333), so protocol failure is not simply the model degrading late in a long exchange.

Violation subtype	All games	Wordle	Taboo
empty / refusal	0	0	0
no format tag	3	0	3
wrong keyword	60	27	33
extra prose	189	146	43
invalid value	35	0	35
Total	**287**	**173**	**114**

G.3 Teacher Demonstrations Repair Participation; Self-ReST Improves Task Performance

In the stage-wise progression base → teacher demonstrations → self-ReST, play-rate is already at ceiling on the three games that work, so every clemscore gain is a quality gain, while the two zero-participation games are never repaired at any stage (Table 19).

G.4 Interactive Improvements Are Not Explained by General Capability Gains

The static, non-interactive probe peaks after demonstrations and then declines under self-ReST even as interactive performance continues to rise, so the interactive gain is not a general capability gain (Table 20).

G.5 Credit-Based Experience Selection Provides a Modest Adaptation Benefit

Under matched compute, credit-weighted selection attains the highest mean clemscore and the lowest spread, but no mean difference against flat rejection sampling survives Holm correction and the variance difference is not significant either (Table 21).

Paired-by-seed comparisons locate the only uncorrected effect in credit weighting over the decoding foil (p=0.032, d_z=0.94); nothing survives Holm correction across the five comparisons (Table 22).

Table 19: **Stage-wise progression, per game (deterministic eval).** Base → +teacher demonstrations → +self-ReST. Play-rate is already at ceiling on the three games that work, and does not move; every clemscore gain is a *quality* gain. The two games at zero participation (Wordle, PrivateShared) are not repaired at any stage — demonstrations do not restore participation where none exists. Teacher demos alone 14.046, +base anchor 14.973 (+0.93), +ReST 20.266 (+5.29).

	Play-rate			Quality			clemscore		
Game	Base	+Demos	+ReST	Base	+Demos	+ReST	Base	+Demos	+ReST
ReferenceGame	1.00	1.00	1.00	33.3	61.1	82.2	33.3	61.1	82.2
ImageGame	1.00	1.00	0.97	0.0	8.2	16.2	0.0	8.2	15.8
Taboo	0.97	0.95	0.43	0.0	5.8	7.7	0.0	5.6	3.3
Wordle	0.00	0.00	0.00	0.0	0.0	0.0	0.0	0.0	0.0
PrivateShared	0.00	0.00	0.00	0.0	0.0	0.0	0.0	0.0	0.0
Overall							**6.667**	**14.973**	**20.266**

Table 20: **Static, non-interactive probe.** Aggregate over the static benchmark suite (MMLU-Pro, BBH, CLadder) for the same three checkpoints.

Checkpoint	Static probe
Base model	30.28
+Teacher demos	35.21
+Self-ReST	33.23

Table 21: **Matched-compute adaptation mechanisms (n=8 seeds, matched across regimes).** SD and CI are recomputed from the stored per-seed clemscores. Credit weighting has the highest mean and the lowest spread, but *no* mean difference against Flat survives correction, and the variance difference is not significant either (Brown–Forsythe flat vs. credit W=1.59, p=0.23; omnibus p=0.58). The one comparison with an uncorrected effect is credit vs. the decoding foil. (matched_n=276).

Method	Mean	SD	95% CI	Δ vs. Flat	d_z
Flat ReST	18.30	1.74	±1.46	—	—
ZPD selection	17.93	1.76	±1.47	-0.37	-0.20
Credit-weighted	19.17	0.88	±0.73	+0.87	+0.43
Decoding foil	17.40	1.64	±1.37	-0.90	-0.32

Table 22: **Pairwise comparisons, paired by seed (n=8).** Uncorrected p, Holm-corrected p across the five comparisons, and paired effect size d_z. Reporting uncorrected p and d_z alongside the correction shows where the signal actually is: credit weighting beats the decoding foil (p=0.03 uncorrected, d_z=0.94) but not flat rejection sampling.

A	B	Δ(A−B)	t	p (unc.)	p (Holm)
ZPD selection	Flat ReST	-0.37	-0.56	0.593	0.788
Credit-weighted	Flat ReST	+0.87	+1.22	0.263	0.788
Decoding foil	Flat ReST	-0.90	-0.92	0.389	0.788
Credit-weighted	ZPD selection	+1.24	+1.94	0.093	0.374
Credit-weighted	Decoding foil	+1.77	+2.67	0.032*	0.161

G.6 Successful Experience Has an Optimal Quantity Rather Than a Monotonic Scaling Law

Self-ReST performance does not scale monotonically with the volume of successful experience: clemscore after one round peaks at N=256 winning trajectories and drops at N=512, though each cell is a single run (Table 23).

Table 23: **Self-ReST does not scale monotonically with the number of winning trajectories.** clemscore after one ReST round trained on N winners. Performance peaks at N=256 and *drops* at N=512, so the binding constraint is not the volume of successful experience. Single run per cell (no seed replication) — we do not read the non-monotonicity as an established effect.

Winners used (N)	64	128	256	512
clemscore	10.90	12.08	18.87	14.16

H Exploratory Analysis: Online Credit Assignment

We report two exploratory on-policy experiments. First, an online GiGPO/GRPO credit loop over the full five-game suite (276 winners per round), in which the model plays, updates immediately, and plays again. Over four rounds performance is non-monotonic and essentially flat (Table 24): 8.15 $\rightarrow$ 9.33 clemscore, far below the 20.27 achieved by offline self-ReST on the same suite. Notably, play-rate *declines* after the first update (0.61 $\rightarrow$ 0.35–0.49), indicating that on-policy training can degrade the protocol competence that scaffolding established. Second, a true single-turn GRPO run (`trl.GRPOTrainer`) on ReferenceGame only, where participation is already saturated (play-rate $= 1.0$): clemscore moves from 34.44 to 36.67, i.e. $+2$ wins out of 90 (31 $\rightarrow$ 33), with almost completely overlapping win-rate CIs ($[0.25, 0.45]$ vs. $[0.27, 0.47]$) — not significant (Table 25). Note this single-game axis is not comparable to the five-game suite axis reported elsewhere. Both results are consistent with the staged account: online credit assignment neither substituted for protocol scaffolding nor, in this regime, improved reasoning beyond offline selection.

Table 24: Exploratory online GiGPO/GRPO credit loop (five-game suite, 276 winners/round). Performance stays flat and below offline self-ReST (20.27), and play-rate drops after the first on-policy update.

Round	Policy	clemscore	Play-rate
0	base (qwen-fmt)	8.15	0.61
1	online-r1	8.56	0.35
2	online-r2	7.78	0.49
3	online-r3	7.33	0.49
4	online-r4	9.33	0.49
Offline self-ReST (ref.)		20.27	—

Table 25: Single-turn GRPO on ReferenceGame (participation already saturated). The gain is $+2$ wins of 90 with overlapping CIs — not significant. Single-game axis; not comparable to suite-level clemscores.

Condition	clemscore	Wins	Play-rate	Win-rate CI
Before (qwen-fmt)	34.44	31/90	1.00	[0.25, 0.45]
After (GRPO)	36.67	33/90	1.00	[0.27, 0.47]