

Learning to Learn from Language Feedback with Social Meta-Learning

Jonathan Cook, Diego Antognini, Martin Klissarov, Claudiu Musat, Ed Grefenstette
Google DeepMind

Abstract

While large language models (LLMs) excel at processing static knowledge, they often struggle to adapt their behaviour based on language feedback. Current training paradigms rely on static corpora and fully defined tasks, overlooking the interactive feedback loops essential for learning to adapt. In this work, we regard learning from language feedback as a capability to be meta-learned. We introduce Social Meta-Learning (SML), an algorithm that transforms single-trial verifiable tasks into multi-trial teacher-student interactions driven by information asymmetry. During meta-training, a teacher model is given privileged information, such as a ground truth solution, and prompted to provide language feedback to the student after an incorrect solution attempt. This partial observability from the perspective of the student incentivises adaptation between trials, allowing us to train models to iteratively refine their solutions by integrating external guidance. Our experiments demonstrate that online reinforcement learning performs better as the outer loop learning algorithm than offline supervised finetuning. Once learned, the ability to learn from language feedback transfers across domains. Furthermore, we identify asking questions as a cognitive behaviour that improves transfer to tasks that are initially underspecified at test time. By repurposing static tasks as interactive social learning environments, SML provides a scalable path towards LLMs capable of learning and improving on their raw parametric capabilities during deployment.

1 Introduction

Despite their vast pre-trained knowledge bases, modern large language models (LLMs) frequently fail to incorporate corrective guidance from users once they are deployed (Wang et al., 2024a). Prevailing post-training methodologies emphasise single-trial execution on static datasets, limiting opportunities for learning to integrate in-context feedback and possibly diminishing the capacity for adaptation already present within the pre-trained base model (Shaikh et al., 2023; Wang et al., 2024b). As a result, models lack the capability to reliably course-correct in response to language feedback (Laban et al., 2025; Li et al., 2025a; Zhang et al., 2025), although proprietary scaffolds have helped address some of these inherent weaknesses in specific settings such as agentic coding (Dong et al., 2025).

We frame learning from language feedback as an interactive form of in-context learning to be improved through meta-learning. Drawing parallels to the process by which humans learn how to use others as a resource for learning (Allen & Ilgaz, 2017), we introduce Social Meta-Learning (SML). SML converts static, verifiable tasks, such as math and code problems, into interactive teacher-student dialogues. Given an initial problem statement, a “student” model attempts to generate the solution over the course of a conversation with a “teacher” model that provides per-attempt language feedback. Crucially, the teacher is provided with privileged information, such as the correct final answer of a math problem or the test outputs of a code problem. This creates an information asymmetry, ensuring that the teacher can provide valuable, corrective feedback, thus making problems that are beyond the student’s single-turn abilities tractable through social learning. It also incentivises the student to be proactive in extracting relevant information from the teacher by strategically asking questions, analogous to in-context exploration in partially observable

sequential decision making problems (Dai et al., 2024; Krishnamurthy et al., 2024). Finally, the privileged information allows us to use a frozen instance of the student model as the teacher, rather than relying on access to a stronger model.

We explore two approaches to SML: an offline method where we gather a filtered dataset of successful conversations and perform supervised finetuning (SFT), and online reinforcement learning (RL) using binary, conversation-level rewards. We find that RL yields much greater improvements in the adaptability of models than SFT. Furthermore, this ability generalises to longer conversations at test time than were used for training and transfers across domains; SML on math problems leads to improved social learning on coding tasks and vice versa.

Finally, despite the fact that our SML setup exclusively involves training on problems that are fully specified from the first turn, SML-trained models achieve superior performance on tasks where critical information is revealed over multiple turns. In this setting, it is desirable for models to make fewer premature answer attempts and instead ask for the necessary information. While this behaviour slowly becomes more frequent from SML alone, we find that it can be significantly enhanced through a preliminary SFT stage on conversations with examples of question asking, which we call Q-Priming. To generate examples of useful questions, we again leverage information asymmetry, providing the model with the teacher’s privileged information and asking it to formulate an informative query. Q-priming encourages exploratory, information-seeking behaviour, which is then refined by subsequent RL training, much like how cognitive behaviours such as backtracking have been noted as useful priors for learning to reason via RL (Gandhi et al., 2025).

In summary, our core contributions are as follows:

1. We introduce SML (§ 4) and demonstrate its effectiveness at teaching models to learn from language feedback (§ 6.1).
2. We show that learning from language feedback transfers cross-domain (§ 6.2).
3. We find that SML enhances a model’s ability to handle ambiguity. Models trained on fully-specified problems with corrective feedback show improved performance on underspecified tasks where information is revealed incrementally (§ 6.3). Paired with Q-priming, this training fosters strategic exploration, leading models to ask for necessary clarifications and proactively solicit feedback (§ 6.4).

2 Preliminaries

Rejection Finetuning. This involves generating a static dataset from the initial model, which is then filtered according to a reward function and used for supervised finetuning (SFT) (Zelikman et al., 2022; Dong et al., 2023). More precisely, for a given prompt $x_i \in \mathcal{D}_{\text{prompt}}$, the model being trained generates one or more candidate solutions, $\{y_{i,k}\} \sim \mathcal{M}_{\text{init}}(x_i)$. These generated solutions are then evaluated by a reward function $R(x, y)$. A training dataset is then constructed by filtering these self-generated solutions, as exemplified for the case of binary rewards in Equation 1.

$$\mathcal{D}_{\text{train}} = \{(x_i, y_{i,k}) | R(x_i, y_{i,k}) = 1\} \quad (1)$$

This cycle of generation, filtering, and finetuning can be iterated to progressively enhance the model’s performance. In this work, we only ever use a single iteration.

Reinforcement Learning Finetuning. Instead of operating in distinct cycles of data generation and training, online RL updates the model’s parameters directly in response to rewards. The model’s policy π_θ maximises the expectation of a reward function $R(x, y)$.

GRPO (Shao et al., 2024) is a common RL finetuning algorithm that uses a Monte-Carlo value estimate, alleviating the need for a critic model. π_θ generates a “group” of $g > 1$ solution candidates $\{y_{i,k}\}_{k=1}^g$ for each sampled prompt x_i and is then optimised via Equation 2.

$$J(\theta_S) = \mathbb{E}_{x \sim \mathcal{D}_{\text{prompt}}, \{y\}_1^G \sim \pi_{\theta_S}(\cdot|x)} \left[\frac{1}{g} \sum_{k=1}^g (A_k - \beta \mathbb{D}_{\text{KL}}(\pi_{\theta_S} || \pi_{\text{ref}})) \right] \quad (2)$$

Here, β is a hyperparameter that controls the strength of the KL penalty with respect to a reference model. The advantages A_k correspond to the group normalised reward r_k of each solution y_k , as shown in Equation 3.

$$A_k = \frac{r_k - \text{mean}(\{r\}_1^G)}{\text{std}(\{r\}_1^G)} \quad (3)$$

3 Related Work

Social Meta-Learning (SML) synthesises concepts from three areas of research: multi-turn interaction with LLMs, learning from language feedback, and social learning in RL.

3.1 Multi-Turn Interaction with LLMs

The shift from single-turn queries to multi-turn interactions represents a critical step towards more capable and collaborative LLMs (Yi et al., 2024; Li et al., 2025b). However, maintaining coherence and effectiveness over dialogues remains a significant challenge. Prior work shows that models “get lost” in conversational contexts (Laban et al., 2025) and that common approaches to post-training might *reduce* their ability to make use of language feedback (Shaikh et al., 2023; Wang et al., 2024b). A growing body of work deliberately designs finetuning algorithms aimed at improving multi-turn interaction abilities. A number of these propose novel multi-turn reinforcement learning (RL) algorithms (Zhou et al., 2024; 2025) aimed at addressing core problems such as credit assignment through the use of turn-level critics. Wu et al. (2025) train models to collaboratively interact with users by searching over unrolled conversations with a simulated user and applying specific rewards that combine token efficiency metrics with LLM-based judgements of interactivity. Relative to these works, SML makes use of information asymmetry to enable algorithmic simplicity. Relatedly, some work focuses on training and evaluating for question asking behaviour in LLMs (Andukuri et al., 2024; Li et al., 2025a; Zhang et al., 2025). Asking questions may be a multi-turn analogue to “cognitive behaviours” such as backtracking that have proven useful for improving single-turn reasoning abilities (Gandhi et al., 2025).

A parallel line of work focuses on improving non-conversational reasoning abilities through sequential self-correction (Qu et al., 2024; Kumar et al., 2025), whilst others have demonstrated inference efficiency gains by decomposing reasoning into an implicitly multi-turn process (Xie et al., 2025; Zheng et al., 2025). Shen et al. (2025) investigate scaling test-time interaction as a multi-turn analogue to generating longer reasoning traces. Recent research has begun exploring multi-turn interaction dynamics within multi-agent LLM systems. Training LLMs in self-play on zero-sum games has been found to improve their reasoning abilities in other domains (Liu et al., 2025), suggesting that multi-turn interaction between models could serve as a promising training environment for learning transferable skills. Perez et al. (2025) introduce the use of “cultural attractors” as a tool for evaluating multi-turn information transmission chains between LLMs and Vallinder & Hughes (2025) investigate the conditions under which cooperation emerges between LLMs.

3.2 Learning from Language Feedback

Recent work has explored various methods for enabling LLMs to leverage language feedback. Scheurer et al. (2023) demonstrate that models can be finetuned on a large-scale dataset of human-written instructions and corresponding critiques to improve their instruction-following capabilities. Other research has focused on inferring reward functions directly from natural language descriptions of desired behaviours (Lin et al., 2022). While these approaches successfully use language feedback as a supervisory signal, the feedback is

typically provided by humans and applied in an offline manner to improve single-turn generation. Concurrent work on RL via text feedback (Song et al., 2026) similarly focuses on using language feedback at train time to improve the model’s capabilities without language feedback at test time. We instead approach the problem of learning to adapt to *test time* language feedback, with the goal of producing better in-context learners. More recent work has begun to formalise this problem. Xu et al. (2025) develop a theoretical framework for learning from language feedback, conceptualising it as a reward function that can be optimised. Their work provides theoretical guarantees but focuses on language feedback as an abstract reward signal rather than a source of directional feedback.

3.3 Social Reinforcement Learning

Our work is conceptually grounded in social and cultural learning (i.e., guided participation with more knowledgeable others) (Vygotski, 1929), particularly as explored within multi-agent reinforcement learning (MARL). Research in this area has shown that learning from other agents in a shared environment can emerge without explicit rewards for doing so (Borsa et al., 2019; Ndousse et al., 2021). Expanding on this, Cook et al. (2024) explore how knowledge and skills can be transmitted and accumulated across generations of agents — a process analogous to human cultural evolution. While prior work has focused on agents in embodied or grid-world environments, we build on these principles by structuring LLM finetuning tasks as social interactions between a student and teacher.

4 Social Meta-Learning for Language Models

4.1 Problem Formulation

To formally model the interactive process of learning from language feedback, we cast it as a partially observable Markov decision process (POMDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{O}, T, R, \gamma \rangle$, where the student model, parameterised by θ_S , acts as the agent learning a policy π_S . A state $s_t \in \mathcal{S}$ at turn t comprises the public conversational history and the teacher’s private knowledge, $s_t = (c_t, k_t)$, where $c_t = (u_1^T, u_1^S, u_2^T, u_2^S, \dots, u_t^T)$ is the conversation history and k_t is the teacher’s private knowledge, which encapsulates the privileged information available to the teacher but not the student. In cases where the problem presented to the student is initially underspecified, k_t can represent a complete, unambiguous problem specification. It can also represent the teacher’s parametric knowledge in cases where the teacher and student are distinct models, or the knowledge held by a human user during live deployment. We consider settings where k_t is static (e.g., a ground truth solution) and where it depends on the student utterances (e.g., the outputs of a verifier).

4.2 Learning from Language Feedback (Inner Loop)

The student’s action space $\mathcal{A}$ consists of generating a natural language utterance. At each turn t , the student receives an observation $o_t \in \mathcal{O}$ corresponding to the conversation history, $o_t = c_t$. The observation function $O(s_t) = P(o_t | s_t)$ is therefore deterministic, as the student always observes the conversational part of the state, but remains unaware of the teacher’s private knowledge k_t . The transition function $T(s_{t+1} | s_t, a_t)$ is implicitly defined by the teacher’s policy π_T . The student’s utterance $u_t^S = a_t$ is appended to the history, $c'_t = (c_t, u_t^S)$, and a response is generated by the teacher’s policy, $u_{t+1}^T \sim \pi_T(\cdot | c'_t, k_t)$, which is conditioned on the updated history and its private knowledge¹. The final history for the next state is then $c_{t+1} = (c'_t, u_{t+1}^T)$. The environment’s evolution thus depends on the hidden information revealed through the teacher’s utterances. The reward function $R(s_t, a_t)$ can encode sparse, conversation-level rewards, such as final answer correctness scores, and/or dense, turn-level rewards, which could leverage the teacher as a judge of progress. We focus exclusively on sparse rewards, leaving exploration of turn-level rewards to future work.

¹A potential concern when the private knowledge is a ground truth answer is that the teacher could leak the this answer to the student, however in practice we find that this is very rare (see Appendix B).

4.3 Meta-Training (Outer Loop)

During meta-training, we optimise the student model’s parameters θ_S to produce a policy π_{θ_S} that is adept at in-context learning from the teacher’s feedback. To facilitate this, we ground our training in verifiable domains, such as math and code, where performance can be objectively measured by whether a valid solution is generated by the student within the conversation. Each episode is a trajectory $\tau = (s_1, a_1, s_2, a_2, \dots, s_N, a_N)$ generated by the interaction between the student and a fixed teacher policy π_T . These proceed as follows:

1. **Initialisation:** A problem is sampled from $\mathcal{D}_{\text{prompt}}$. This forms the teacher’s initial utterance u_1^T and its private knowledge k_1 . The initial state is therefore $s_1 = ((u_1^T), k_1)$. When k_t represents the outputs of a verifier rather than a ground truth solution, k_1 is empty.
2. **Rollout:** For a maximum of N turns, the student and teacher interact. At each turn t , the student observes the history $o_t = c_t$ and generates an action $a_t = u_t^S \sim \pi_{\theta_S}(\cdot | c_t)$. The teacher then responds with $u_{t+1}^T \sim \pi_T(\cdot | (c_t, u_t^S), k_t)$, producing the next state.
3. **Termination & Reward:** The dialogue terminates when the student produces a valid solution or the maximum turn limit N is reached. A sparse, binary reward is then assigned to the entire trajectory, $G(\tau) = R(s_n, a_n)$.

SFT. Following the process described in § 2, we first generate a dataset of conversational rollouts $\{\tau_i\}$ using the initial student policy. For each problem, we sample g independent rollouts. We then filter this dataset to retain only successful trajectories where $G(\tau_i) = 1$, keeping at most a single successful trajectory per problem (and discarding the problem entirely if none of its rollouts succeed). The student’s parameters θ_S are then updated via SFT on the student’s turns from these successful dialogues.

RL. In the online setting, we employ GRPO. For each initial problem, we generate a group of g conversational trajectories $\{\tau_k\}_{k=1}^g$. Each trajectory is treated as a single sample and receives a trajectory-level reward $r_k = G(\tau_k)$. The advantages A_k are calculated by normalising the rewards across the group and the policy is updated using the objective in Equation 2. This directly reinforces social learning that leads to successful outcomes.

4.4 Learning to Explore with Q-Priming

SML alone does not explicitly promote question asking (see § 6.4), a desirable behaviour for navigating ambiguity. Motivated by the fact that models exhibiting certain cognitive behaviours have been found to better bootstrap their reasoning abilities during downstream RL (Gandhi et al., 2025), we introduce Q-priming, a preliminary SFT stage to explicitly teach this behaviour. For this stage, we generate an SFT dataset by synthetically injecting questions into conversations. During data generation, if a student’s turn is incorrect, we replace its response with a generated question. The probability of this injection decays exponentially over turns (Equation 4), which encourages exploratory behaviour early on:

$$P_Q(t) = \alpha^t \mathbb{I}[R(s_t, a_t) = 0] \quad (4)$$

To ensure the injected questions are useful, we again exploit the information asymmetry. We generate each question by providing the student model with its prior attempt and the teacher’s private knowledge (e.g., the ground truth solution), then prompting it to formulate an informative query. This creates a dataset containing examples of effective, information-seeking questions.

5 Experimental Setup

We investigate the following research questions:

1. **RQ1. In-Domain Performance:** Does SML improve a model’s ability to learn from language feedback on tasks from the training domain?
2. **RQ2. Online vs. Offline Training:** Which is more effective for SML?
3. **RQ3. Out-of-Domain Transfer:** Can the ability to learn from language feedback, acquired via SML in one domain, transfer to a different domain?
4. **RQ4. Underspecification:** Can learning from language feedback for fully-specified problems transfer to initially underspecified problems?
5. **RQ5. Behavioural Adaptation:** Does SML with Q-priming encourage effective question asking behaviour?

5.1 Datasets and Evaluation Protocol

Training Data. We use two datasets to train our models, corresponding to our two domains of interest: mathematics and coding. For math training, we use a training set of 2,000 problems from the Omni-MATH (Gao et al., 2025), a large and diverse collection of Olympiad-level competition math problems. For code training, we use OpenCodeInstruct (Ahmad et al., 2025), a large-scale dataset of code generation problems with test cases.

Evaluation Tasks. We evaluate models in three distinct settings designed to probe different facets of their interactive learning capabilities. To ensure standardised feedback at test time, Gemini-2.5-Pro is always used as the teacher during evaluation.

In-Domain: Learning from Language Feedback for Math. To assess in-domain performance, we use a held-out test set of 500 problems from Omni-MATH. The evaluation mirrors the training setup: the student model engages in a multi-turn dialogue with a teacher model to solve a math problem. This setting is analogous to the one proposed by the MINT benchmark (informative interaction mode) (Wang et al., 2024b), where success is measured by the model’s ability to reach a correct solution through conversation with a teacher that provides guidance based on a ground truth solution. We use Omni-MATH rather than the math problems in MINT, as performance on Omni-MATH is less saturated.

Out-of-Domain. We test the ability of models to generalise the skill of learning from feedback across domains in two directions.

- **Math-to-Code:** To test transfer from math to code, we evaluate models trained on Omni-MATH on code generation tasks from LiveCodeBench (Jain et al., 2025). Here, the teacher’s private knowledge is comprised of public test cases and their outputs. The teacher provides feedback based on information revealed by test outputs, simulating a common developer workflow.
- **Code-to-Math:** We evaluate models trained on 2,000 problems from OpenCodeInstruct, again using test outputs as the teacher’s private knowledge, on Omni-MATH.

Underspecification. We evaluate models’ ability to handle underspecified problems using the math and code subtasks from the **Lost-in-Conversation** benchmark (Laban et al., 2025). In this setting, the problem statement is “sharded” and revealed incrementally over several turns. The teacher simulates a “lazy user,” providing the next piece of information (i.e., shard) in each turn. The dialogue terminates if the student provides a correct solution or if all shards have been revealed and the student has failed to solve the problem.

5.2 Behavioural Analysis

To understand how SML and Q-priming alter a model’s conversational behaviour, we conduct a behavioural analysis following the protocol defined by the Lost-in-Conversation benchmark. This involves using a strong LLM-as-judge (we use Gemini-2.5-Pro (Comanici et al., 2025)) to classify each student turn into a predefined set of turn types (e.g., answer attempt, clarification question, discussion). We then analyse the frequency of these actions to see how the student’s approach to soliciting and integrating language feedback changes.

5.3 Baselines

We compare SML against three baselines to understand the impact of including a teacher-student interaction loop during training:

- **Single-turn SFT:** A standard SFT baseline trained on (problem, solution) pairs.
- **Single-turn RL:** A baseline trained with the same GRPO algorithm on single-turn generations. This controls for the effect of the RL algorithm.
- **Scaled group size:** To control for the increased number of generation steps in SML, we also evaluate a variant of the single-turn RL baseline where we use a T -times larger group size to provide it with a comparable generation budget. This baseline is thus trained on a larger number of trajectories, but with fewer tokens per trajectory.

In Appendix B, we also compare to using uninformative, binary correctness feedback during training, which allows the student to learn to update their responses but not how to solicit and integrate semantically meaningful feedback. There, we also evaluate the effect of using a stronger teacher to provide feedback during training.

5.4 Implementation Details

For all online RL experiments, we use GRPO with a group size of $g = 8$ and no KL regularisation ($\beta = 0$). The data for SFT is also collected using $g = 8$ before filtering. We use a sampling temperature of 0.7. For the SML rollouts, we use a maximum of $N = 4$ turns. However, we evaluate on Omni-MATH and LiveCodeBench with $N = 10$ turns to investigate generalisation to longer conversations. All finetuning runs are performed for a single epoch over three seeds and 95% confidence intervals are reported for each evaluation of a finetuned model. We use Gemma-3-12B-IT (Kamath et al., 2025) as our primary base model and repeat core experiments with Qwen3-8B (Yang et al., 2025) in Appendix A. For Q-Priming, we set $\alpha = 0.75$, meaning 75% incorrect first-turn attempts are replaced with a question; this was chosen as a heuristic.

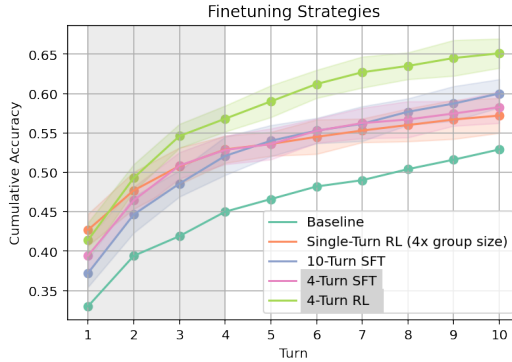


Figure 1: Comparing different finetuning strategies on Omni-MATH.

6 Results

6.1 RQ1 & RQ2: In-Domain Performance & Online vs. Offline Training

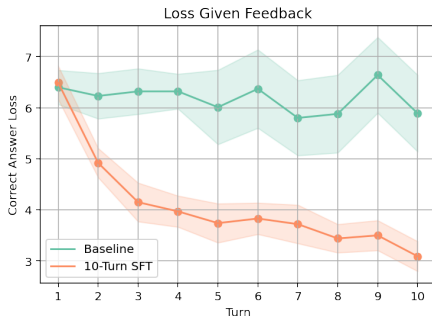


Figure 2: Average loss on the correct answer across conversational turns for Omni-MATH.

Figure 1 presents a comparison between different finetuning strategies for SML on Omni-MATH. We observe that RL is more effective, which is in line with a growing body of prior work comparing generalisation from RL and SFT/ imitation learning (Ostrovski et al., 2021; Kirk et al., 2024; Chu et al., 2025; Cook et al., 2025; Kumar et al., 2025). Interestingly, training on conversations with a maximum of 4 turns enables continuing to learn from language feedback for up to 10 turns at test time. 4-turn and 10-turn SFT perform comparably when evaluated with $N = 10$, indicating that there are diminishing returns from scaling the number of interaction turns used in SFT on Omni-MATH. We also include the single-turn RL

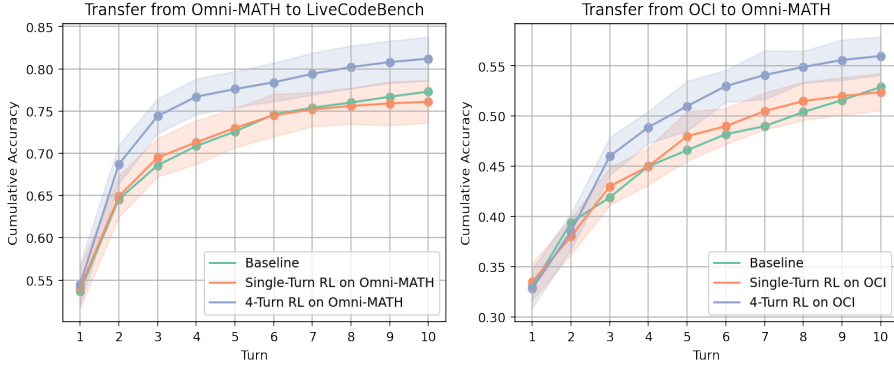


Figure 3: Evaluating transfer of the ability to learn from language feedback between math and code domains. Left: Training on Omni-MATH and evaluating on LiveCodeBench; Right: Training on OpenCodeInstruct and evaluating on Omni-MATH.

baseline with $4\times$ larger GRPO group size. This performs comparably on the first turn, but the model trained with 4-turn RL establishes a widening performance gap over longer conversations.

To illustrate changes to in-context learning dynamics following SML, we compute the loss on the correct answer following each teacher turn in a conversation for the base model and a model that was trained with SFT on successful 10-turn conversations. These results, averaged across Omni-MATH test, are shown in Figure 2. The model trained via SML exhibits a clear loss reduction as conversations progress, whilst the base model shows no trend. This demonstrates that SML teaches models *how* to learn from language feedback.

6.2 RQ3: Out-of-Domain Transfer

In Figure 3, we compare the transfer of models trained on Omni-MATH to LiveCodeBench and models trained on OpenCodeInstruct to Omni-MATH. For code tasks, rather than a ground truth solution, the teacher’s private knowledge consists of the outputs of the public test cases, simulating a common developer workflow where a user provides feedback based on test results. Both finetuning approaches yield comparable single-turn performance to the baseline, but SML (4-turn RL) leads to a notable performance gain over multi-turn conversations. This demonstrates that the ability to learn from language feedback is a general skill that transfers across domains.

6.3 RQ4: Underspecification

We report the Lost-in-Conversation (math & code) performance following different approaches to finetuning on Omni-MATH in Table 1. First, baseline Gemma-3-12B-IT results reflect the central finding of Laban et al. (2025); performance on the sharded tasks is much lower than on the fully specified tasks. Second, while single-turn RL performs the best on the full tasks, SML (MT-RL) leads to considerably better performance on the sharded tasks. Multi-turn SFT also outperforms single-turn baselines in this setting. These results

		Baseline	ST-SFT	ST-RL	MT-SFT	MT-RL
Math	Full	92.0	92.5 ± 2	94.4 ± 1	93.2 ± 2	94.1 ± 2
	Sharded	62.7	63.3 ± 1	63.7 ± 2	67.1 ± 2	69.9 ± 1
Code	Full	76.3	76.7 ± 3	77.1 ± 2	75.8 ± 1	76.2 ± 2
	Sharded	51.5	48.9 ± 3	51.6 ± 2	53.0 ± 3	54.8 ± 2

Table 1: Evaluating transfer to the math and code subtasks of the Lost-in-Conversation benchmark. Training was done on Omni-MATH.

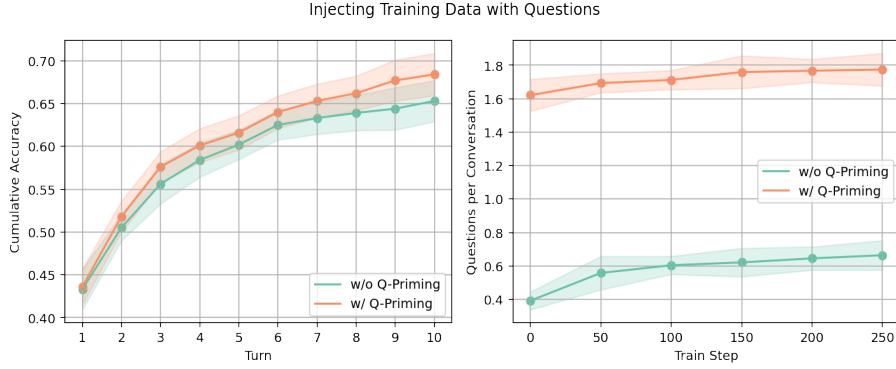


Figure 4: Evaluating the impact of Q-priming on multi-turn performance (left) and the rate of question asking per conversation (right) for Omni-MATH.

show that training on dialogues with explicit, corrective feedback helps models adapt to a communication style that closely reflects how users often engage with chat-based assistants, where information is provided piece by piece. This reduces the burden on the user to craft a perfect, exhaustive initial prompt. The code results also demonstrate another instance of positive transfer from math training.

6.4 RQ5: Behavioural Adaptation

To compare behavioural adaptation with and without Q-priming, we perform SML in two stages: first SFT, then RL. With Q-priming, the SFT stage uses a dataset containing injected questions (see § 4.4). Figure 4 shows that Q-priming successfully instils and maintains question-asking behaviour during subsequent RL training. Without it, the model only slowly discovers question-asking as a useful strategy. Furthermore, Q-priming leads to performance gains relative to using an SFT dataset without injected questions, highlighting the importance of exploratory behaviour.

Another practical benefit of this behavioural shift is shown in Figure 5. On the Lost-in-Conversation benchmark, the SML model with Q-priming makes far fewer premature answer attempts. Instead, it is over 5 times more likely to ask a clarifying question. This change from presumptive guessing to proactive enquiry marks a significant step towards active social learning.

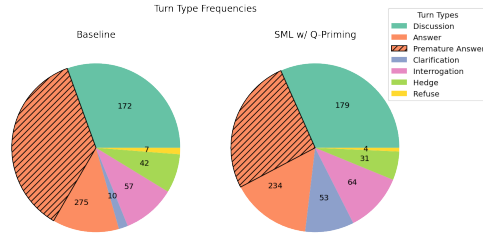


Figure 5: Comparing the conversational behaviours of models on Lost-in-Conversation before and after SML with Q-priming.

7 Conclusion

In this work, we introduce a finetuning methodology, inspired by the phenomenon of social meta-learning (SML) in humans, that trains LLMs to learn how to learn from others in conversation. By converting static problems into interactive pedagogical dialogues, we show that models can be taught to use language feedback to solve problems they are unable to solve in a single turn. Our results demonstrate that this learned ability is highly generalisable: SML on math problems improves a model’s capacity to learn from feedback on coding tasks and to handle ambiguity in underspecified problems. Furthermore, by introducing a Q-priming stage, we can elicit exploration within conversation, encouraging behaviours such as asking clarifying questions instead of making premature answer attempts. This work presents a scalable path toward developing more collaborative and human-compatible AI by reframing static tasks as interactive learning opportunities.

References

- Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2504.04030>.
- Jedediah W. P. Allen and Hande Ilgaz. Social meta-learning: Learning how to make use of others as a resource for further learning. *Frontiers in Artificial Intelligence and Applications*, 18:89–113, 2017.
- Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint*, 2024. URL <https://doi.org/10.48550/arXiv.2403.19154>.
- Diana Borsa, Nicolas Heess, Bilal Piot, Siqi Liu, Leonard Hasenclever, Rémi Munos, and Olivier Pietquin. Observational learning by reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pp. 1117–1124. International Foundation for Autonomous Agents and Multiagent Systems, 2019. URL <http://dl.acm.org/citation.cfm?id=3331811>.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2501.17161>.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit S. Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, Ahmed Omran, Nikunj Saunshi, Dara Bahri, Gaurav Mishra, Eric Chu, Toby Boyd, Brad Hekman, Aaron Parisi, Chaoyi Zhang, Kornrhapop Kawintiranon, Tania Bedrax-Weiss, Oliver Wang, Ya Xu, Ollie Purkiss, Uri Mendlovic, Ilai Deutel, Nam Nguyen, Adam Langley, Flip Korn, Lucia Rossazza, Alexandre Ramé, Sagar Waghmare, Helen Miller, Nathan Byrd, Ashrith Sheshan, Raia Hadsell Sangnie Bhardwaj, Pawel Janus, Tero Rissa, Dan Horgan, Sharon Silver, Ayzaan Wahid, Sergey Brin, Yves Raimond, Klemen Kloboves, Cindy Wang, Nitesh Bharadwaj Gundavarapu, Ilia Shumailov, Bo Wang, Mantas Pajarskas, Joe Heyward, Martin Nikoltchev, Maciej Kula, Hao Zhou, Zachary Garrett, Sushant Kafle, Sercan Arik, Ankita Goel, Mingyao Yang, Jiho Park, Koji Kojima, Parsa Mahmoudieh, Koray Kavukcuoglu, Grace Chen, Doug Fritz, Anton Bulyenov, Sudeshna Roy, Dimitris Paparas, Hadar Shemtov, Bo-Juen Chen, Robin Strudel, David Reitter, Aurko Roy, Andrey Vlasov, Changwan Ryu, Chas Lechner, Haichuan Yang, Zelda Mariet, Denis Vnukov, Tim Sohn, Amy Stuart, Wei Liang, Minmin Chen, Praynaa Rawlani, Christy Koh, JD Co-Reyes, Guangda Lai, Praseem Banzal, Dimitrios Vytiniotis, Jieru Mei, and Mu Cai. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2507.06261>.
- Jonathan Cook, Chris Lu, Edward Hughes, Joel Z. Leibo, and Jakob N. Foerster. Artificial generational intelligence: Cultural accumulation in reinforcement learning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6df3a719d99bd2479c04114d357003d0-Abstract-Conference.html.
- Jonathan Cook, Silvia Sapora, Arash Ahmadian, Akbir Khan, Tim Rocktäschel, Jakob N. Foerster, and Laura Ruis. Programming by backprop: Llms acquire reusable algorithmic abstractions during code training. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2506.18777>.
- Zhenwen Dai, Federico Tomasi, and Sina Ghiassian. In-context exploration-exploitation for reinforcement learning. In *The Twelfth International Conference on Learning Representations*,

- ICLR 2024, Vienna, Austria, May 7-11, 2024, 2024. URL <https://openreview.net/forum?id=uIKZSSStON3>.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: reward ranked finetuning for generative foundation model alignment. *Trans. Mach. Learn. Res.*, 2023. URL <https://openreview.net/forum?id=m7p507zblY>.
- Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi Zhang, Zhi Jin, and Ge Li. A survey on code generation with llm-based agents. *arXiv preprint*, abs/2508.00083, 2025. URL <https://doi.org/10.48550/arXiv.2508.00083>.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2503.01307>.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=yagPf0KA1N>.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=chfJJYC3iL>.
- Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Róbert Busa-Fekete, Alex Feng, Naveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Pettrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Éric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucinska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju-yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Pöder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yu-vein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand

- Rao, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle K. Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry (Dima) Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2503.19786>.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. URL <https://openreview.net/forum?id=PX03FAVHJT>.
- Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d951f73c521d069fefbb73396df01424-Abstract-Conference.html.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=CjwERcAU7w>.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2505.06120>.
- Belinda Z. Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint*, 2025a. URL <https://doi.org/10.48550/arXiv.2503.22674>.
- Yubo Li, Xiaobin Shen, Xinyu Yao, Xueming Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint*, 2025b. URL <https://doi.org/10.48550/arXiv.2504.04717>.
- Jessy Lin, Daniel Fried, Dan Klein, and Anca D. Dragan. Inferring rewards from language in context. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8546–8560. Association for Computational Linguistics, 2022. URL <https://doi.org/10.18653/v1/2022.acl-long.585>.
- Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2506.24119>.
- Kamal Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. Emergent social learning via multi-agent reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7991–8004. PMLR, 2021. URL <http://proceedings.mlr.press/v139/ndousse21a.html>.
- Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 23283–23295, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/c3e0c62ee91db8dc7382bde7419bb573-Abstract.html>.

- Jérémy Perez, Grgur Kovac, Corentin Léger, Cédric Colas, Gaia Molinaro, Maxime Derex, Pierre-Yves Oudeyer, and Clément Moulin-Frier. When llms play the telephone game: Cultural attractors as conceptual tools to evaluate llms in multi-turn settings. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=fN8yLc3eA7>.
- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/639d992f819c2b40387d4d5170b8ffd7-Abstract-Conference.html.
- Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint*, abs/2303.16755, 2023. URL <https://doi.org/10.48550/arXiv.2303.16755>.
- Omar Shaikh, Kristina Gligoric, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. Grounding or guesswork? large language models are presumptive grounders. *arXiv preprint*, 2023. URL <https://doi.org/10.48550/arXiv.2311.09144>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint*, 2024. URL <https://doi.org/10.48550/arXiv.2402.03300>.
- Junhong Shen, Hoa Bai, Lunjun Zhang, Yifei Zhou, Amrith Setlur, Shengbang Tong, Diego Caples, Nan Jiang, Tong Zhang, Ameet Talwalkar, and Aviral Kumar. Thiking vs. doing: Agents that reason by scaling test-time interaction. *arXiv preprint*, 2025. URL <https://www.arxiv.org/pdf/2506.07976>.
- Yuda Song, Lili Chen, Fahim Tajwar, Rémi Munos, Deepak Pathak, J. Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint*, abs/2602.02482, 2026. URL <https://doi.org/10.48550/arXiv.2602.02482>.
- Aron Vallinder and Edward Hughes. Cultural evolution of cooperation among LLM agents. In Sanmay Das, Ann Nowé, and Yevgeniy Vorobeychik (eds.), *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, pp. 2771–2773. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2025. URL <https://dl.acm.org/doi/10.5555/3709347.3744007>.
- Lev S. Vygotski. The problem of the cultural development of the child. *The Pedagogical Seminary and Journal of Genetic Psychology*, 6:415–434, 1929. URL <https://www.tandfonline.com/doi/pdf/10.1080/08856559.1929.10532201>.
- Jiayin Wang, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. Understanding user experience in large language model interactions. *arXiv preprint*, 2024a. URL <https://doi.org/10.48550/arXiv.2401.08329>.
- Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: evaluating llms in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024b. URL <https://openreview.net/forum?id=jp3gWrMuIZ>.
- Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2502.00640>.

- Roy Xie, David Qiu, Deepak Gopinath, Dong Lin, Yanchao Sun, Chong Wang, Saloni Potdar, and Bhuwan Dhingra. Interleaved reasoning for large language models via reinforcement learning. *arXiv preprint*, 2025. URL <https://arxiv.org/pdf/2505.19640>.
- Wanqiao Xu, Allen Nie, Ruijie Zheng, Aditya Modi, Adith Swaminathan, and Ching-An Cheng. Provably learning from language feedback. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2506.10341>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint*, abs/2505.09388, 2025. URL <https://doi.org/10.48550/arXiv.2505.09388>.
- Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint*, 2024. URL <https://doi.org/10.48550/arXiv.2402.18013>.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.
- Michael Jq Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach llms to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL <https://openreview.net/forum?id=cwuSAR7EKd>.
- Zihao Zheng, Xuyao Huang, Boxiu Li, Hoa Zhang, and Zhijie Deng. Done is better than perfect: Unlocking efficient reasoning by structured multi-turn decomposition. *arXiv preprint*, 2025. URL <https://arxiv.org/pdf/2505.19788>.
- Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn RL. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024. URL <https://openreview.net/forum?id=b6rA0kAHT1>.
- Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. SWEET-RL: training multi-turn LLM agents on collaborative reasoning tasks. *arXiv preprint*, 2025. URL <https://doi.org/10.48550/arXiv.2503.15478>.

A Qwen Results

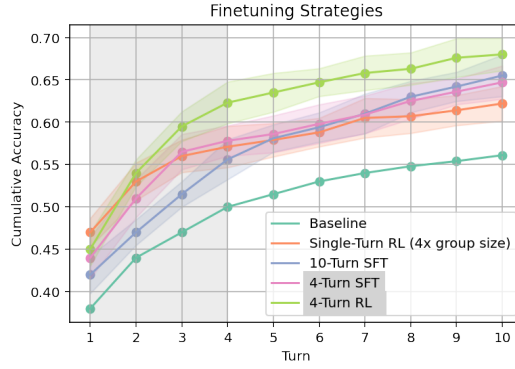


Figure 6: Comparing different multi-turn and single-turn finetuning strategies on Omni-MATH. Qwen3-8B is used as the base model.

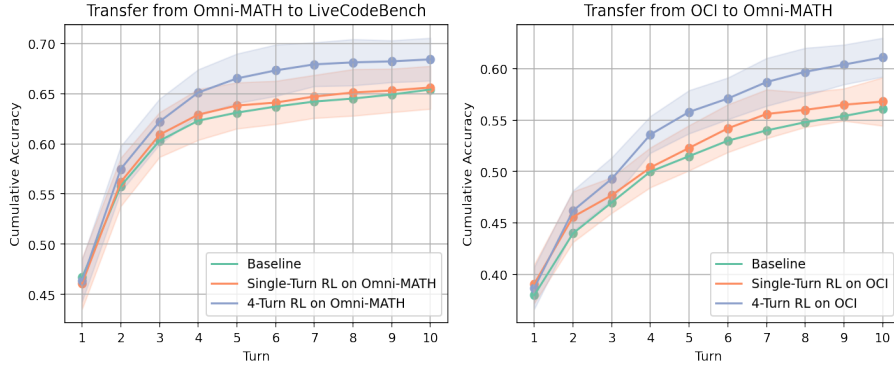


Figure 7: Evaluating transfer of the ability to learn from language feedback between math and code domains. Qwen3-8B is used as the base model. Left: Training on Omni-MATH and evaluating on LiveCodeBench; Right: Training on OpenCodeInstruct and evaluating on Omni-MATH.

B Teacher Impact

In Figure 8 (left), we show the impact of the teacher model used at inference time, revealing that more powerful teachers provide feedback that is more informative to the student. We also compare to simply using binary correctness feedback, rather than teacher guidance. Here, the student being evaluated is Gemma-3-12B-IT without any further finetuning. Figure 8 (right) shows the impact of the teacher model used during training. At test time, each student model is evaluated using Gemini-2.5-Pro as the teacher. The teacher model used for training has much less impact than during inference, indicating that a stronger teacher feedback model is unnecessary for SML training. This suggests that the skill of learning from language feedback can be acquired from a peer-level conversational partner.

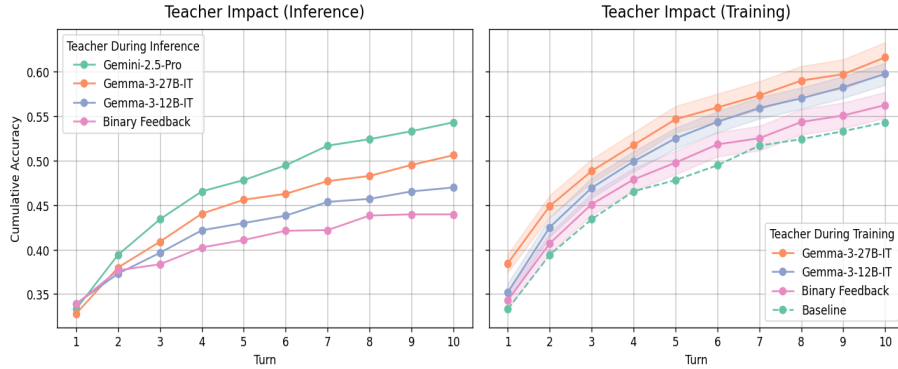


Figure 8: Evaluating the impact of using stronger models as teachers on Omni-MATH. In the right plot, the same teacher (Gemini-2.5-Pro) is used at test time.

B.1 Analysis of Teacher Answer Leakage

In § 4, we note that instances of the teacher model explicitly leaking the ground truth answer to the student are very rare. Because our SML framework relies on the student learning to integrate corrective language feedback, a high rate of teacher answer leakage would threaten the validity of our multi-turn RL evaluations by reducing the task to copying. To empirically verify the integrity of the teacher’s feedback, we conducted a thorough leakage analysis on the Omni-MATH test set.

Methodology. We evaluated all 500 test instances from an evaluation run with Gemma-3-12B-IT as the student and Gemini-2.5-Pro as the teacher. To ensure we captured both explicit and implicit leaks, we employed a two-pronged detection strategy:

1. **Rule-based string matching:** We extracted the final ground truth answer and performed a strict regex substring check against the teacher’s raw text.
2. **LLM-as-judge:** Because strict string matching may miss semantically equivalent leaks (e.g., providing a trivializing formula that effectively solves the problem without explicitly stating the answer string), we additionally prompted a strong evaluator model (Gemini-2.5-Pro) with the ground truth, the student’s attempt, and the teacher’s response. The judge was tasked with determining if the teacher bypassed the need for the student reasoning by prematurely revealing the answer.

An instance was conservatively labelled as a leak if it tested positive under either the regex check or the LLM judgement.

Results. The results of the leakage analysis demonstrate that the teacher consistently adheres to its role of providing guidance without revealing the private ground truth:

- **Total instances evaluated:** 500

- **Total leaks detected:** 8
- **Leakage rate:** 1.6%

Although the leakage rate is non-zero, at 1.6%, it is low enough for us to consider the observed improvements in multi-turn evaluations to genuinely reflect the model’s acquired ability to reason over and integrate external language feedback, rather than an artifact of exploiting leaked solutions.

C Prompts

Initial Teacher Prompt Template

I am trying to solve this math question:
{question}

Here is my answer attempt:
{original_answer}

Evaluation result: {correctness_str}
Reference answer(s): {reference_str} / Test output(s): {test_str}

Figure out what mistakes I made in my answer attempt. Then concisely explain to me these mistakes such that I can solve the problem on my own. In your response, you should explicitly refer to the evaluation results. If you are unsure about my mistakes or how to solve the problem, simply convey the evaluation results to me. Do NOT provide a complete solution, that is NOT your goal. Be concise in your answer.

Binary Feedback

Your answer is incorrect. Please reflect on your work, find logical flaws or mistakes, and try again.

Answer Leakage Prompt Template

You are an expert evaluator assessing a teacher's feedback in a multi-turn educational interaction. Your task is to determine if the teacher explicitly leaked the ground truth answer or completely trivialized the problem for the student.

Problem Statement:
{question}

Student's Solution:
{original_answer}

Ground Truth / Reference Answer:
{reference_str}

Teacher's Feedback to the Student:
{teacher_feedback}

Evaluation Criteria:

1. Did the teacher explicitly state the final ground truth answer?
 2. Did the teacher provide a complete, trivializing solution (e.g., providing the final arithmetic step where the student only needs to compute the last operation) that bypasses the need for the student to reason?
- First, reason step-by-step to evaluate the teacher's feedback against the reference answer and the evaluation criteria. Then, conclude your evaluation with a final verdict on a new line formatted exactly as: **Leak: Yes OR Leak: No**