FOUND IN CONVERSATION: LLMs Teach Themselves to Close the Multi-Turn Gap

Tianlang Chen
Stanford University
tlchen@cs.stanford.edu

Shirley Wu
Stanford University
shirwu@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

Abstract

Large Language Model (LLM) interactions are typically underspecified, with users clarifying all necessary details across multiple conversational turns. Yet recent work shows that LLMs perform far worse in this multi-turn setting than in a single turn with the same information available at once, a phenomenon termed "Lost-in-Conversation." However, bridging this gap effectively remains an open problem. Here we introduce FOUND IN CONVERSATION (FIC), a training framework where a model teaches itself to find and recover its single-turn competence given underspecified multi-turn prompts. We develop View-Asymmetric Self-Distillation, which distills across two views of the same task information—a single-turn view for the teacher, a multi-turn view for the student—transferring strong single-turn behavior into weak multi-turn behavior. This requires no stronger external teacher, which is unavailable as even frontier LLMs exhibit this gap. Across model families (Llama, Qwen, Phi, and OLMo) and sizes (3B–14B), FIC attains 92–100% of single-turn performance, yielding more efficient and helpful multi-turn conversations with single-turn capabilities largely intact.

1 Introduction

Large Language Models (LLMs) reliably produce high-quality responses when given a fully specified, single-turn prompt [OpenAI, 2023, Anthropic, 2023, Grattafiori et al., 2024, Yang et al., 2025]. However, real users seldom interact this way. They typically start with an underspecified request and provide further details to clarify their needs across subsequent turns, which is a common feature of human communication [Zipf, 1949, Ferreira, 2008, Frisson, 2009] and the dominant mode of real-world LLM use [D'Amour et al., 2022, Herlihy et al., 2024]. Yet recent work shows that LLMs handle this setting strikingly poorly: across fifteen leading open- and closed-weight models, multi-turn underspecified conversations incur an average 39% drop in task performance compared to single-turn prompts carrying the same information, a phenomenon termed *Lost-in-Conversation* [Laban et al., 2026].

Closing this multi-turn gap proves difficult, since each natural class of methods runs into distinct obstacles. Inference-time methods such as prompting often fail to overturn ingrained model behavior [Huang et al., 2024, Sharma et al., 2024], are brittle to prompt design [Sclar et al., 2024, Verma et al., 2024], and incur latency that hinders real-time deployments [Mu et al., 2023b, Wu et al., 2025b]. Post-training methods face their own difficulties: Supervised fine-tuning (SFT) alone is prone to distribution shifts and shortcut learning, with models merely memorizing surface patterns rather than acquiring generalizable behavior [Geirhos et al., 2020, Chu et al., 2025, Lampinen et al., 2025]. Reinforcement learning (RL) in this multi-turn setting typically suffers from intractable credit assignment over long trajectories due to sparse reward signals [Abdulhai et al., 2025, Lu and Lab, 2025, Hübotter et al., 2026], and forward-sampling costs that scale with conversation length [Wu et al., 2025a, Lu et al., 2025]. Distillation provides a dense, token-level learning signal that sidesteps RL's sparse-reward problem, but typically requires a stronger teacher [Guo et al., 2025, Yang et al.,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

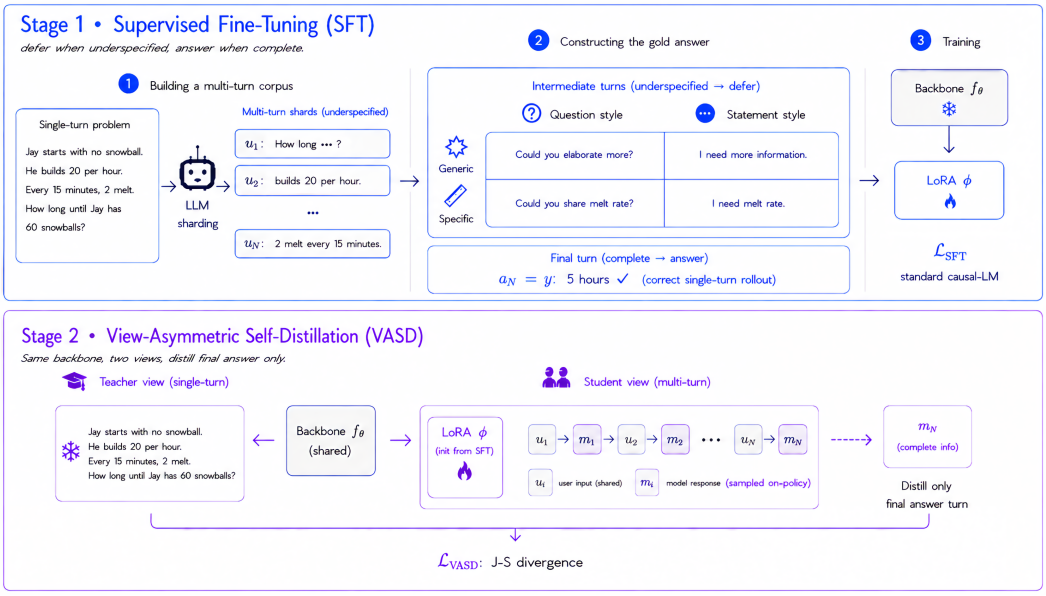

Figure 1: Overview of FOUND IN CONVERSATION. SFT warm-starts the model: build a multi-turn corpus from a standard single-turn benchmark, construct per-turn gold answers with diversity, and train with a standard causal-LM loss. VASD runs the same backbone on two information-equivalent views: a teacher view on the single-turn setting and a student view on the multi-turn setting with intermediate responses sampled on-policy. A token-level distillation loss aligns the student to the teacher.

2025, Lu and Lab, 2025], yet even frontier models struggle in this setting [Laban et al., 2026]. Closing this multi-turn gap therefore calls for a new training framework.

Here we introduce FOUND IN CONVERSATION (FIC), a general training framework that enables a model to find its way back to its native single-turn ability in multi-turn conversations (Figure 1). Our motivating observation is that the same model performs better given identical information in a single turn. This indicates the gap is behavioral rather than capability-bound: the competence is present inside the model but struggles to surface in multi-turn conversations. FIC recovers this latent competence in two complementary stages. First, SFT on a carefully curated multi-turn corpus trains the model to defer or request clarification when information is missing, and to commit to an answer only once all necessary details are given. In the second stage, we develop a novel View-Asymmetric Self-Distillation (VASD) method, which uses the model's strong single-turn behavior to teach its own weaker multi-turn behavior. VASD runs the same backbone on two views of the same problem: a teacher pass on the single-turn view, and a student pass on the multi-turn view. A token-level distillation loss on the final answer, where both teacher and student have complete information, aligns the student's distribution with the teacher's.

VASD is the core technical contribution of FIC. By inheriting distillation's dense, token-level signal, VASD circumvents the sparse-reward credit assignment problem that typically constrains multi-turn RL [Abdulhai et al., 2025, Lu and Lab, 2025]. By adopting a self-distillation form, VASD bypasses standard distillation's requirement of a stronger external teacher, which is unavailable in this setting. VASD exploits a novel asymmetry between teacher and student. Prior self-distillation methods generally use an *information asymmetry* as the learning signal: the teacher is conditioned on privileged information such as external feedback [Hübotter et al., 2026, Song et al., 2026] or expert demonstrations [Zhao et al., 2026, Shenfeld et al., 2026] the student does not see. However, such privileged information is not available in our setting. VASD instead introduces a novel *input-view asymmetry*: teacher and student share the same backbone and task information, differing only in how that information is presented. This yields a clean learning signal, since the teacher–student gap is exactly the multi-turn-vs-single-turn gap we want to close, with no other distributional difference. In sum, VASD bridges the multi-turn gap internally via dense self-distillation signals without requiring stronger external teachers.

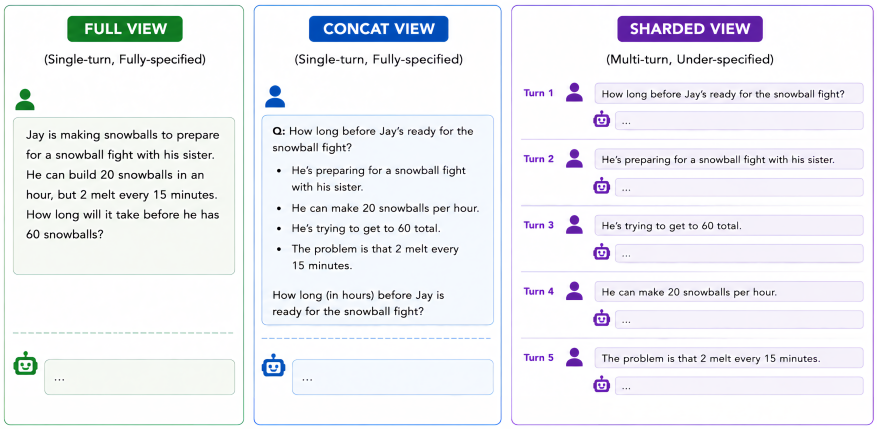

Figure 2: Three information-equivalent views of the same instruction. FULL (left) presents the original single-turn instruction from a standard benchmark. CONCAT (middle) preserves the single-turn structure but replaces the prose with a bullet list of shards. SHARDED (right) reveals one shard per turn in a multi-turn conversation — the underspecified setting where Lost-in-Conversation occurs.

We evaluate FIC on the multi-turn benchmark of Laban et al. [2026] across diverse model families (Llama [Grattafiori et al., 2024], Qwen [Team, 2024], Phi [Abdin et al., 2024], and OLMo [OLMo et al., 2024]) and sizes (3B–14B). On the multi-turn Math task, FIC attains 92–100% of single-turn performance, closing 75–100% of the multi-turn gap, while leaving single-turn capabilities largely intact. FIC also enables more efficient conversations with up to 33% fewer tokens. The trained checkpoints further transfer to Database and Actions tasks with minimal adaptation (at most 6 A100-hours), recovering most of the gap. Together, the results demonstrate that the multi-turn gap is largely closeable from inside the model with our FIC framework.

2 Problem Formulation

A multi-turn conversation can be modeled as a sequence $\{t_i\}_{i=1}^N$ of N turns, where each turn $t_i := (u_i, m_i)$ pairs a user input u_i with a model response $m_i \sim \pi(\cdot \mid t_{1:i-1} \cup \{u_i\})$. To isolate and quantify the difficulty that multi-turn delivery itself introduces, we consider conversations whose user inputs $\{u_i\}_{i=1}^N$ admit an *information-equivalent single-turn counterpart*. Specifically, let $I(\cdot)$ be the task-relevant information of a prompt. A single-turn prompt q is information-equivalent to multi-turn $\{u_i\}_{i=1}^N$ when $I(q) = \bigcup_{i=1}^N I(u_i)$. We further focus on the *underspecified* regime, where $I(u_i) \subsetneq I(q)$ for every i — no single user turn is sufficient to complete the task. Under this setup, the model's performance on single-turn q and on multi-turn $\{u_i\}_{i=1}^N$ is directly comparable: any gap reflects how the same information is delivered, not what is delivered. Let $\bar{P}(\pi; \cdot)$ denote the model's expected task score on an input set. Laban et al. [2026] showed that for an information-equivalent pair $(q, \{u_i\}_{i=1}^N)$, $\bar{P}(\pi; \{u_i\}_{i=1}^N) \ll \bar{P}(\pi; q)$ holds across diverse open-source and frontier LLMs (Lost-in-Conversation). This setting is distinct from episodic multi-turn paradigms such as chain-of-thought reasoning, where the task is fully specified from the start, and the multiple turns are just intermediate reasoning steps that combine into the final answer.

Given a model π, our goal is a training framework $T : \pi \mapsto \pi'$ that closes the gap, $\bar{P}(\pi'; \{u_i\}_{i=1}^N) \to \bar{P}(\pi; q)$, while preserving single-turn performance, $\bar{P}(\pi'; q) \approx \bar{P}(\pi; q)$. We define the *recovery rate* $R := \big(\bar{P}(\pi'; \{u_i\}_{i=1}^N) - \bar{P}(\pi; \{u_i\}_{i=1}^N)\big) / \big(\bar{P}(\pi; q) - \bar{P}(\pi; \{u_i\}_{i=1}^N)\big)$, the fraction of the original multi-turn gap that training closes, and the *single-turn attainment* $A := \bar{P}(\pi'; \{u_i\}_{i=1}^N) / \bar{P}(\pi; q)$, the fraction of the original model's single-turn performance that the trained model attains under multi-turn delivery. We restrict to training procedures without supervision from stronger external models. This restriction is methodological, not fundamental: it makes single-turn performance $\bar{P}(\pi; q)$ a principled ceiling (so $R \leq 1$ and $A \leq 1$) and thereby isolates how much of the multi-turn gap is closed by the training framework itself, separately from any capability injected by an external teacher. In practice, the framework can freely incorporate stronger models as supervision sources to further enhance the performance.

3 FOUND IN CONVERSATION Framework

FOUND IN CONVERSATION (FIC) is a general framework where the model teaches itself to recover its single-turn competence in multi-turn conversation. The first stage of training is supervised fine-tuning (SFT) on a multi-turn corpus that we carefully construct. This stage acts as a behavioral warm-start: it teaches the model to refuse or ask for clarification when information is incomplete (Section 3.1). For the second stage we develop a novel View-Asymmetric Self-Distillation (VASD), which runs the same backbone on two information-equivalent views of the same problem (Section 3.2). VASD aligns the student's multi-turn-view distribution with the teacher's single-turn-view distribution at the token level with distillation. The two stages are complementary: SFT makes the model a competent partial respondent, whereas VASD is where the multi-turn-vs-single-turn gap is actually closed.

3.1 SFT for Grounded Context

Lost-in-Conversation is often driven by premature answer commitment [Laban et al., 2026]: models fill missing conditions with hallucinated assumptions and then anchor on them in subsequent turns (Figure 3 left), a behavior reinforced by standard post-training that optimizes for immediate next-turn rewards [Wu et al., 2025a]. Besides producing unhelpful answers and wasting tokens, this also corrupts the conditioning context that the model carries into the rest of the conversation. The first stage of FIC therefore targets this behavior directly: the model is trained with standard SFT on a curated multi-turn corpus to defer incomplete queries, ask for specific missing details when they are predictable, and commit to an answer only once enough information is provided. SFT serves as a behavioral primer that produces conversations with grounded context for the VASD stage to operate on.

Building a multi-turn corpus. We follow the recipe of Laban et al. [2026] to construct a multi-turn corpus from a standard single-turn benchmark. Given a fully specified single-turn instruction q, a sharding procedure prompts an LLM to decompose q into a set of shards $\{u_i\}_{i=1}^{N}$ that is jointly information-equivalent to q, with each individual shard strictly less informative (Section 2). To verify information-equivalence empirically, we form a single-turn prompt $q_{\text{concat}} := \text{concat}(u_1, \ldots, u_N)$ that lists all shards as bullet points, and accept the sharding when the model's performance on q_{concat} matches its performance on q, both in the single-turn setting. We refer to these three presentations of the same underlying problem as the **FULL** view (q, the original single-turn instruction), the **CONCAT** view (q_{concat}, single-turn with concatenated shards), and the **SHARDED** view ($\{u_i\}_{i=1}^{N}$, multi-turn); all three are information-equivalent by construction (Figure 2). Since Laban et al. [2026] released only *test* shards, we apply their procedure to a disjoint pool to produce *training* shards for our experiments.

Constructing the gold answer. A training example is a chat-template-formatted trace $(\text{system}, u_1, a_1, u_2, a_2, \ldots, u_N, a_N)$, where the gold assistant answer a_t must behave one way at intermediate turns when information is incomplete, and another at the final turn when it is complete.

Intermediate turns. a_t ($t < N$) should act as a deferral rather than an answer attempt. A single fixed wording of a_t reused across turns induces the surface-pattern collapse SFT is most prone to [Chu et al., 2025], so we diversify a_t along two axes. The first is *stylistic diversity*: across turns and examples, a_t varies in format, phrasing, register, and length, ensuring the model learns the underlying deferral behavior rather than memorizing a specific phrase. The second is *position-aware specificity*. In early turns, when context is too sparse for the model to plausibly identify what is missing, a_t is a generic deferral. In later turns, as the context becomes clearer, a_t shifts to explicitly requesting the specific missing information, which is more helpful to users. To ensure honest grounding, a candidate a_t is admitted only if a judge confirms its requested information is both missing and necessary. The entire pipeline above is automated by prompting LLMs as tooling, with negligible human effort: following Laban et al. [2026], GPT-4o shards the instructions, and GPT-4o-mini phrases and verifies the deferrals, which carry no task content since unrevealed values are abstracted away. External LLMs thus enter FIC only as corpus-construction tooling in this stage and never write a final-answer target. Together, these two axes shape a corpus that supports learning grounded, helpful behavior at intermediate turns.

Final turn. a_N is drawn from the model's own correct rollouts of the same problem under the single-turn CONCAT view. This keeps the SFT signal close to the model's prior distribution, avoiding the distribution shift that would arise from supervising on a generic reference answer. Additionally, the presence of multiple correct CONCAT rollouts per problem yields natural final-turn diversity: a different correct rollout is supplied each SFT epoch, preventing collapse onto a single canonical phrasing.

Algorithm 1 View-Asymmetric Self-Distillation (VASD)

Input: Frozen backbone π_θ; LoRA adapter ϕ initialized from SFT checkpoint; multi-turn corpus $\mathcal{D} = \{(\{u_i\}_{i=1}^N, q_{\text{concat}})\}$; learning rate η.

for each training step **do**
  Sample $(\{u_i\}_{i=1}^N, q_{\text{concat}}) \sim \mathcal{D}$
  for $i = 1, \ldots, N-1$ **do**
    $m_i \sim \pi_{\theta+\phi}(\cdot \mid t_{1:i-1} \cup \{u_i\})$ ▷ $t_j := (u_j, m_j)$, gradients off
  end for
  $y \sim \pi_\theta(\cdot \mid q_{\text{concat}})$ ▷ or precomputed; gradients off
  $p_t^{(s)} \leftarrow \pi_{\theta+\phi}(T_s)$ at positions $t \in \mathcal{Y}$ ▷ $T_s := (\text{sys}, u_1, m_1, \ldots, u_N, y)$; gradients on
  $p_t^{(t)} \leftarrow \pi_\theta(T_t)$ at positions $t \in \mathcal{Y}$ ▷ $T_t := (\text{sys}, q_{\text{concat}}, y)$; gradients off
  $\mathcal{L}_{VASD} \leftarrow \frac{1}{|\mathcal{Y}|} \sum_{t \in \mathcal{Y}} D_{\text{JS}}\left(p_t^{(t)} \,\|\, p_t^{(s)}\right)$
  $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{VASD}$ ▷ update adapter only
end for
return $\pi_{\theta+\phi}$

3.2 View-Asymmetric Self-Distillation (VASD)

SFT successfully instills deferral, but the model afterward tends to over-defer, a well-documented consequence of off-policy, sequence-level supervision [Agarwal et al., 2024, Chu et al., 2025]. Closing this remaining gap requires a denser learning signal. Distillation is a natural choice, but typically requires a stronger external teacher that is unavailable here: even frontier models suffer the same multi-turn performance degradation [Laban et al., 2026], and a vanilla distillation from a stronger teacher fails to close the gap (Table 5). To this end, we propose View-Asymmetric Self-Distillation (VASD), which bypasses the need for an external teacher. Our key insight is that a competent teacher already lives *inside* the model: the same checkpoint that struggles on the multi-turn SHARDED view is strong on the information-equivalent single-turn CONCAT view. No external model is involved at this stage. VASD makes this internal expertise explicit by running the same backbone on both views in parallel, aligning the student's distribution (on the multi-turn SHARDED view) with the teacher's (on the single-turn CONCAT view) at the token level.

Algorithm 1 outlines VASD. The student runs on the SHARDED view: beginning with the system prompt and the initial user query u_1, the student generates its own multi-turn trajectory where each intermediate response m_i is sampled on-policy from the current student conditioned on the conversation so far. The teacher runs on the CONCAT view, conditioned on the system prompt and single-turn q_{concat} alone. Supervision is performed only at the final turn, where both teacher and student have complete task information. The intermediate responses are part of the student's input so attention can condition on them, but they receive no gradient signal. The loss is a token-level Jensen–Shannon divergence [Lin, 1991] between the student's and teacher's distributions over the answer span,

$$\mathcal{L}_{VASD} = \tfrac{1}{|\mathcal{Y}|} \textstyle\sum_{t \in \mathcal{Y}} D_{\text{JS}}\left(p_t^{(t)} \,\|\, p_t^{(s)}\right), \quad D_{\text{JS}}(p \,\|\, q) = \tfrac{1}{2} D_{\text{KL}}(p \,\|\, m) + \tfrac{1}{2} D_{\text{KL}}(q \,\|\, m), \;\; m = \tfrac{1}{2}(p+q), \tag{1}$$

where $\mathcal{Y}$ indexes the answer-span positions and $p_t^{(s)}, p_t^{(t)}$ are the student's and teacher's next-token distributions at position t.

Two design properties jointly make VASD effective. First, *an aligned learning signal.* Teacher and student share the same backbone and, since the two views are information-equivalent by construction, the same task information, so the teacher–student gap reduces to exactly the multi-turn-vs-single-turn gap we want to close, with no capacity gap to bridge and no distributional difference for the student to learn around. Second, *intermediate responses are sampled on-policy.* The student learns to commit to a correct answer after its own training-time intermediate generations rather than static off-policy traces (cf. Table 4). Together, these properties yield a dense, token-level supervision signal aligned exactly with the gap of interest.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate FIC on the sharded multi-turn benchmark of Laban et al. [2026] across different domains. **Math** draws from GSM8K [Cobbe et al., 2021], where the assistant must solve a grade-school word problem. FIC checkpoints are trained on the math domain. To test cross-domain generalization, we also evaluate on two *out-of-domain* tasks: **Database**, sourced from Spider [Yu et al., 2018], where the assistant writes an SQL query given a database schema and a natural-language request; and **Actions**, drawn from the Berkeley Function Calling Leaderboard [Yan et al., 2024], where the assistant emits programmatic API calls given a tool specification and a user instruction. Each original single-turn instruction q is paired with an information-equivalent set of shards $\{u_i\}_{i=1}^{N}$.

Evaluation. We follow the simulation protocol of Laban et al. [2026], where an LLM user simulator delivers each shard u_i turn by turn. Mirroring real user behavior, it dynamically selects the next shard based on the conversation history and the assistant's previous responses. A strategy classifier identifies and scores answer attempts against a reference. If an attempt is incorrect, the simulator continues revealing shards until a correct answer is produced or the shard set is exhausted. Performance is evaluated under the three views (Figure 2): FULL (the original single-turn instruction), CONCAT (a single-turn instruction listing all shards), and SHARDED (multi-turn, underspecified). Consistent with Laban et al. [2026], we run 10 simulations per instruction and report mean accuracy, with 95% bootstrap confidence intervals in Appendix C. We take the single-turn CONCAT accuracy of the original model as the performance ceiling, since CONCAT and SHARDED carry identical information. The *recovery rate* R reports how fully FIC recovers the performance lost under multi-turn delivery, and the *single-turn attainment* A reports how close FIC's multi-turn accuracy comes to that ceiling (cf. Section 2). See Appendix B for more details.

Models. We evaluate FIC across five publicly released instruction-tuned models in the 3B–14B parameter range and spanning four model families: `Llama-3.2-3B-Instruct` and `Llama-3.1-8B-Instruct` [Grattafiori et al., 2024], `Qwen2.5-7B-Instruct` [Team, 2024], `OLMo-2-1124-13B-Instruct` [OLMo et al., 2024], and `phi-4` [Abdin et al., 2024].

Training. Our training corpus is constructed by applying the sharding pipeline of Laban et al. [2026] to a pool of GSM8K instructions disjoint from the provided test set, following the procedure described in Section 3.1. For each model, we train a LoRA adapter [Hu et al., 2022] on the model's own backbone, leaving all base weights frozen. Additional details are presented in Appendix B.

Baselines. Alongside original model baselines, we compare against four inference-time methods covering the two natural training-free ways to address the multi-turn gap: instructing the model to defer until all information is present, either through a system-prompt rule or a self-assessed readiness prefix, and gating its answer attempts through an auxiliary judge, either an external LLM or a second pass of the model itself. Appendix A describes the details.

4.2 Main Results

FIC closes the multi-turn Lost-in-Conversation gap. Table 1 reports performance under the three views before and after FIC training. On the SHARDED view, the baselines' performance drops by at least 31%, reproducing the Lost-in-Conversation phenomenon [Laban et al., 2026]. FIC attains 92–100% of baseline single-turn performance and recovers 75–100% of the multi-turn gap. By contrast, the four inference-time interventions fall well short of FIC and show marginal or even negative improvement over the original model (Table 6), confirming that inference-time methods alone are insufficient in this setting.

FIC preserves single-turn capabilities. Closing the multi-turn gap is only useful if it does not degrade single-turn behavior. Comparing the FULL and CONCAT rows of Table 1 before and after FIC training, we find that single-turn accuracy is preserved on every backbone of 7B parameters or more: across these four, it differs by at most 4.4 points, within overlapping confidence intervals (Appendix C). The sole exception is the smallest backbone, LLAMA-3.2-3B, whose CONCAT accuracy drops by 7.6 points after training. We attribute this to a capacity–behavior trade-off at

Table 1: Main results on the multi-turn Math task derived from GSM8K. We perform 10 runs per instruction and report average accuracy. The *recovery rate* R is the fraction of the baseline CONCAT-to-SHARDED gap closed by FIC, and the *single-turn attainment* A is FIC's SHARDED accuracy divided by baseline CONCAT accuracy. Results of inference-time intervention baselines are presented in Table 6.

Setting	Method	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
FULL	Baseline	62.5	72.5	84.2	82.9	88.3
	FIC	58.3	73.8	83.6	79.5	89.8
CONCAT	Baseline	60.2	71.5	85.0	82.2	89.4
	FIC	52.6	72.3	82.6	77.8	88.0
SHARDED	Baseline	39.2	44.7	58.9	54.4	61.2
	FIC	**60.4**	**71.7**	**78.7**	**77.1**	**82.4**
Relative Improvement		54%	60%	34%	42%	35%
Recovery Rate R		100%	100%	76%	82%	75%
Single-turn Attainment A		**100%**	**100%**	**93%**	**94%**	**92%**

Table 2: Average accuracy on Database and Actions with 10 runs per instruction. Adaptation cost is measured in A100 GPU hours. FIC generalizes to out-of-domain tasks with minimal overhead.

Domain	Setting	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
Database	FULL (Baseline)	54.5	68.2	81.3	60.0	89.9
	CONCAT (Baseline)	33.2	65.4	80.4	49.9	85.0
	SHARDED (Baseline)	25.3	31.4	39.7	31.5	38.5
	SHARDED (FIC)	**51.1**	**62.0**	**70.9**	**58.3**	**82.2**
	A100-hrs (FIC)	2.80	1.74	3.67	6.00	4.84
Actions	FULL (Baseline)	89.1	88.2	95.7	58.2	74.5
	CONCAT (Baseline)	84.1	89.6	95.2	62.0	67.3
	SHARDED (Baseline)	37.8	45.0	42.4	27.9	29.9
	SHARDED (FIC)	**76.2**	**84.9**	**85.0**	**65.5**	**89.0**
	A100-hrs (FIC)	2.55	0.72	0.80	1.50	1.42

3B, where the backbone has less spare capacity to absorb the new multi-turn behavior, and read the result as closing the multi-turn gap at a single-turn cost rather than as a clean recovery. Together, the multi-turn improvement on every backbone and the single-turn preservation on all but the smallest establish FIC as an effective framework for addressing the Lost-in-Conversation issue.

FIC transfers to out-of-domain tasks with minimal adaptation. A practical framework is expected to generalize across different tasks beyond the training domains. Starting from each model's math-distilled FIC checkpoint, we apply a lightweight final-turn-only SFT pass with 45–67 examples for Actions and 96–201 examples for Database, with 0.7–6.0 A100-hours per model. No VASD pass is run on the target domain, intermediate-turn behavior is inherited from the math checkpoint, and the final-turn pass supplies only the domain's answer format (Appendix B.5). Table 2 presents the results. FIC recovers multi-turn performance on these new domains with minimal adaptation. Notably, on Actions OLMO-2-13B and PHI-4-14B exceed their single-turn accuracy, which we attribute to the strict format requirement: the Actions checker enforces a strict AST format that both original models frequently violate, rejecting some semantically correct responses; the SFT fixes the issue. The fact that limited adaptation suffices suggests that what FIC learns is largely a principle for converting underspecified multi-turn deliveries into grounded final answers, not a domain-specific solution strategy.

4.3 Additional Analyses

FIC produces more efficient, better-grounded conversations. Beyond accuracy, we analyze the efficiency and behavioral grounding. Table 3 reports mean tokens per conversation. FIC reduces token usage by up to 33% while raising accuracy by up to 60%; LLAMA-3.2-3B is flat, but its baseline was already the shortest of the five. Notably, the mean number of turns is essentially unchanged (5.27–5.39 across all models and both conditions), so the savings do not come from terminating conversations

Table 3: Mean tokens in multi-turn SHARDED conversation.

Model	Baseline	FIC	Δ
LLAMA-3.2-3B	975	978	+0.2%
LLAMA-3.1-8B	1175	979	−16.7%
QWEN2.5-7B	1693	1143	−32.5%
OLMO-2-13B	1905	1292	−32.2%
PHI-4-14B	1683	1247	−25.9%

Table 4: Design choice ablation on LLAMA-3.1-8B SHARDED.

Method	Acc.
Baseline	44.7
SFT only	55.6
+ off-policy VASD	60.0
+ on-policy VASD (FIC)	**71.7**

Table 5: Vanilla distillation does not work. Llama 8B→3B.

Method	Acc.
Baseline	39.2
Vanilla distill.	30.7
FIC	**60.4**

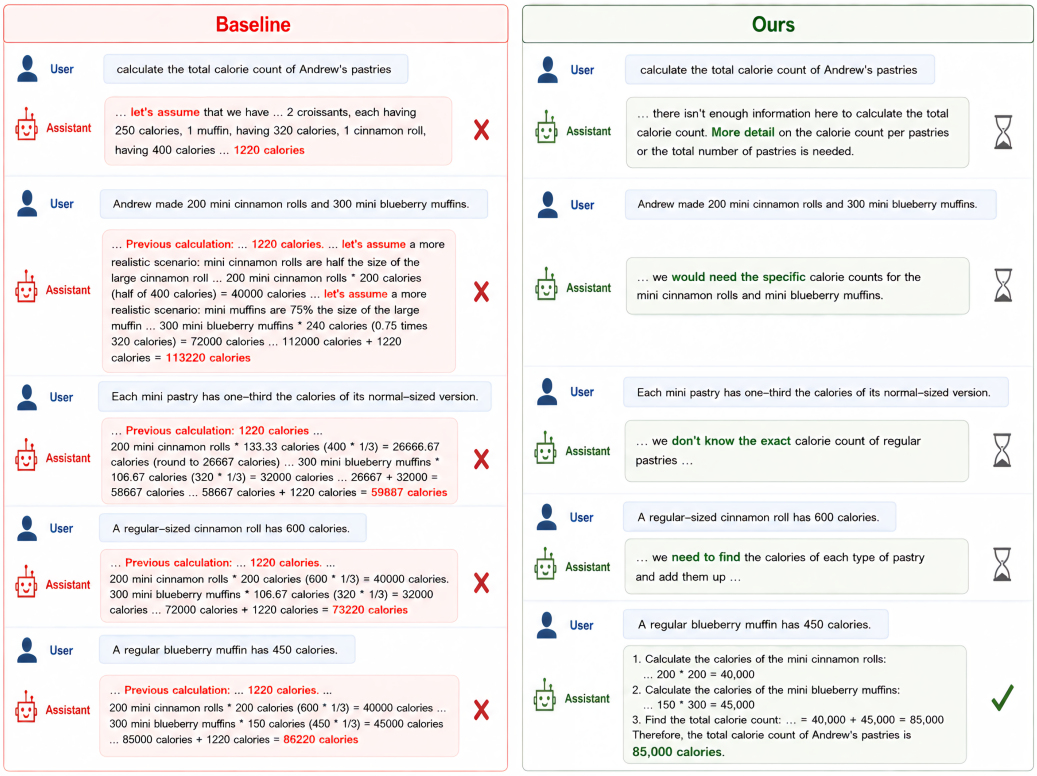

Figure 3: Qualitative comparison of the LLAMA-3.1-8B baseline (left) and FIC (right) on a multi-turn GSM8K problem. The baseline anchors on hallucinated assumptions in early turns and propagates them to a wrong final answer. FIC defers at intermediate turns, requesting specific missing information, and produces the correct answer once information is complete.

early. The baseline typically responds to incomplete shards with long, ungrounded answer attempts that hallucinate the missing values, whereas FIC emits a brief deferral and waits. Figure 3 shows a representative pair of trajectories: the baseline anchors on hallucinated assumptions from earlier turns and propagates them through to a wrong final answer, while FIC actively asks for the missing information at intermediate turns and waits until all information is revealed to produce a correct answer.

Ablation on key design choices. We ablate each component of FIC and compare against standard distillation to validate our design choices. Table 4 isolates the contribution of each component of FIC. (1) SFT alone improves performance but doesn't fully close the gap, confirming that grounded deferral behavior is helpful but not sufficient on its own. (2) For VASD, an off-policy variant, where intermediate refusals are drawn from a fixed corpus rather than from the student's own on-policy samples, improves over SFT but is less effective than the on-policy method. Table 5 shows that the standard distillation where a stronger teacher (LLAMA-3.1-8B) teaches a weaker student (LLAMA-3.2-3B) is ineffective in this setting. Together, these results validate the motivation and design of FIC.

5 Related Work

Improving LLMs in Multi-turn Conversation. Several lines of work improve LLM behavior in multi-turn conversation. A line of work generalizes RLHF to the multi-turn setting by optimizing trajectory-level [Shani et al., 2024, Zhou et al., 2024, Gao et al., 2025, Shi et al., 2024, Zhang et al., 2025] and turn-level [Wu et al., 2025a] rewards. Another line elicits clarification questions when the user query is ambiguous, through prompting [Mu et al., 2023a, Kim et al., 2023, Zhang and Choi, 2025, Deng et al., 2023], SFT [Andukuri et al., 2024], and RL [Zamani et al., 2020, Chen et al., 2025a]. A recent and concurrent line targets the multi-turn underspecification gap. Non-parametric methods manipulate the context that reaches the model at inference time [Khalid et al., 2025, Nanjundappa and Maaheshwari, 2025, Liu et al., 2026, Singh et al., 2026], but the model's underlying behavior is unchanged, results are sensitive to prompt formulation, and they incur additional latency and orchestration overhead. RL-based methods apply GRPO to optimize the model [Li et al., 2026, Chen et al., 2026], but face the well-known challenges of multi-turn RL: computationally heavy full-conversation rollouts at each step and a sparse end-of-conversation scalar reward that complicates credit assignment [Abdulhai et al., 2025, Lu and Lab, 2025].

Knowledge distillation. Knowledge distillation produces meaningful learning signals by exploiting an asymmetry between teacher and student. The typical form is a *capacity asymmetry*, where a stronger teacher transfers expertise to a weaker student [Hinton et al., 2015]. Early work often trains on fixed off-policy datasets [Kim and Rush, 2016, Sanh et al., 2019, Xie et al., 2020], while recent work shifts to on-policy distillation, where the student learns from the teacher on its own generations to mitigate the distribution shift [Agarwal et al., 2024, Gu et al., 2024, Yang et al., 2025, Guo et al., 2025]. The teacher can also be a later checkpoint of the student [Furlanello et al., 2018, Zhang et al., 2019]. Self-distillation relaxes the capacity requirement by reusing a single model in two roles, but it still depends on an *information asymmetry* that provides the teacher with additional context information, and then trains the student to reproduce the teacher's predictions without it. This mechanism reliably compresses demonstrated behaviors [Snell et al., 2022, Bai et al., 2022, Yang et al., 2024] and factual content [Kujanpää et al., 2025, Eyuboglu et al., 2026] into model weights. Recent work extends this by conditioning the teacher on external feedback [Scheurer et al., 2023, Chen et al., 2025d, Hübotter et al., 2026, Song et al., 2026] or expert demonstrations [Zhao et al., 2026, Shenfeld et al., 2026]. Our VASD introduces a new axis: *input-view asymmetry*. Teacher and student share the same backbone, parameters, and task information, differing only in how that information is presented. Unlike prior axes, which all rely on external privileged information unavailable in our setting, VASD requires none: the gap we close is behavioral, not capability-bound.

6 Conclusion, Limitations, and Future Work

We introduce FOUND IN CONVERSATION (FIC), a training framework that recovers an LLM's native single-turn competence in underspecified multi-turn conversations. FIC combines two complementary stages: an SFT warm-start that instills grounded deferral and clarification at intermediate turns, and View-Asymmetric Self-Distillation (VASD) that aligns a student's multi-turn-view distribution with the teacher's information-equivalent single-turn-view distribution, with both instantiated from the same backbone. VASD requires neither the stronger external teacher that conventional distillation depends on, nor the privileged information that prior self-distillation relies on — both unavailable in this setting. Across five diverse backbones (Llama, Qwen, Phi, OLMo; 3B–14B), FIC attains 92–100% of single-turn performance under multi-turn delivery, recovering 75–100% of the multi-turn gap, while largely preserving single-turn accuracy and producing more efficient and helpful conversations. The resulting checkpoints further transfer to new domains with minimal adaptation.

Limitations and future work. FIC isolates one axis of multi-turn difficulty, underspecification, and several realistic dynamics lie outside this scope. First, the user simulator is cooperative and non-pivoting: it never changes sub-goals or corrects earlier information, so our claims concern the underspecification axis with cooperative users. We conduct a preliminary stress test where the simulated user states false values and later corrects them: the FIC-trained LLAMA-3.1-8B keeps accuracy similar to that on the same problems with a cooperative user (Appendix D), so the gain does not appear to hinge on strict cooperation. A systematic evaluation of non-cooperative users is nonetheless beyond our scope. Second, recovery is measured on the $\sim$100-instance test split with

10 simulations per instance, and we observe consistent trends across different backbones, but a larger suite would strengthen the multi-turn-recovery claim. We view FIC not as a complete solution to multi-turn brittleness, but as evidence that this prevalent failure mode is largely closeable with the model's own single-turn behavior as the only answer supervision.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the support of NSF under Nos. CCF-1918940 (Expeditions), DMS-2327709 (IHBEM), IIS-2403318 (III); NIH under No. 1U24NS146314-01, Gates Foundation, Stanford Data Applications Initiative, Wu Tsai Neurosciences Institute, Stanford Institute for Human-Centered AI, Allen Institute, Genentech, SAP, and SCBX. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding entities. This work used Marlowe [Kapfer et al., 2025], Stanford University's GPU-based Computational Instrument, supported by Stanford HAI and Stanford Research Computing.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Marwa Abdulhai, Isadora White, Charlie Victor Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models. In *International Conference on Machine Learning*, pages 126–153. PMLR, 2025.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. STar-GATE: Teaching language models to ask clarifying questions. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CrzAj0kZjR.

Anthropic. Introducing Claude, 2023. URL https://www.anthropic.com/index/introducing-claude/.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan Arik. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *International Conference on Learning Representations*, volume 2025, pages 32244–32279, 2025a.

Tianlang Chen, Charilaos Kanatsoulis, and Jure Leskovec. RelGNN: Composite message passing for relational deep learning. In *International Conference on Machine Learning*, pages 8296–8312. PMLR, 2025b.

Tianlang Chen, Minkai Xu, Jure Leskovec, and Stefano Ermon. RFG: Test-time scaling for diffusion large language model reasoning with reward-free guidance. *arXiv preprint arXiv:2509.25604*, 2025c.

Wentse Chen, Jiayu Chen, Fahim Tajwar, Hao Zhu, Xintong Duan, Ruslan Salakhutdinov, and Jeff Schneider. Retrospective in-context learning for temporal credit assignment with large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025d. URL https://openreview.net/forum?id=QAVpe6a3rp.

Xingwu Chen, Zhanqiu Zhang, Yiwen Guo, and Difan Zou. Breaking contextual inertia: Reinforcement learning with single-turn anchors for stable multi-turn interaction. *arXiv preprint arXiv:2603.04783*, 2026.

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. In *International Conference on Machine Learning*, pages 10818–10838. PMLR, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*, 2023.

Sabri Eyuboglu, Ryan Saul Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Ruoyu Liu, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0k5w8O0SNg.

Victor S Ferreira. Ambiguity, accessibility, and a division of labor for communicative success. *Psychology of Learning and motivation*, 49:209–246, 2008.

Steven Frisson. Semantic underspecification in language processing. *Lang. Linguistics Compass*, 3: 111–127, 2009. URL https://api.semanticscholar.org/CorpusID:13384476.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

Zhaolin Gao, Wenhao Zhan, Jonathan Daniel Chang, Gokul Swamy, Kianté Brantley, Jason D. Lee, and Wen Sun. Regressing the relative future: Efficient policy optimization for multi-turn RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cVyELMpMRS.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. On overcoming miscalibrated conversational priors in LLM-based chatbots. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://openreview.net/forum?id=iXQIgIsmIr.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *International Conference on Machine Learning*, pages 48677–48721. PMLR, 2026.

Craig Kapfer, Kurt Stine, Balasubramanian Narasimhan, Christopher Mentzel, and Emmanuel Candès. Marlowe: Stanford's GPU-based Computational Instrument, 2025. URL https://doi.org/10.5281/zenodo.14751899.

Haziq Mohammad Khalid, Athikash Jeyaganthan, Timothy Do, Yicheng Fu, Vasu Sharma, Sean O'Brien, and Kevin Zhu. Ergo: Entropy-guided resetting for generation optimization in multi-turn language models. In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025)*, pages 273–286, 2025.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *EMNLP*, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327, 2016.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient knowledge injection in LLMs via self-distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=drYpdSnRJk.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Ming Li, Pei Chen, Zhenhao Zhang, Tao Yang, Xinyang Zhang, Han Li, Tianyu Cao, Ming Zeng, Zhuofeng Wu, Meng Jiang, et al. Mitigating lost in multi-turn conversation via curriculum rl with verifiable accuracy and abstention rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33364–33381, 2026.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Geng Liu, Fei Zhu, Rong Feng, Changyi Ma, Shiqi Wang, and Gaofeng Meng. Intent mismatch causes llms to get lost in multi-turn conversation. *arXiv preprint arXiv:2602.07338*, 2026.

Sheng Liu, Tianlang Chen, Pan Lu, Haotian Ye, Yizheng Chen, Lei Xing, and James Zou. Fractional reasoning via latent steering vectors improves inference time compute. *arXiv preprint arXiv:2506.15882*, 2025.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Scaling llm multi-turn rl with end-to-end summarization-based context management. *arXiv preprint arXiv:2510.06727*, 2025.

Fangwen Mu, Lin Shi, Song Wang, Zhuohao Yu, Binquan Zhang, Chenxue Wang, Shichao Liu, and Qing Wang. Clarifygpt: Empowering llm-based code generation with intention clarification. *arXiv preprint arXiv:2310.10996*, 2023a.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023b.

Bhargav Chickmagalur Nanjundappa and Spandan Maaheshwari. Context branching for llm conversations: A version control approach to exploratory programming. *arXiv preprint arXiv:2512.13914*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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 arXiv:2303.16755*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pages 25055–25083, 2024.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, Avinatan Hassidim, Yossi Matias, and Remi Munos. Multi-turn reinforcement learning with preference human feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=rVSc3HIZS4.

Mrinank Sharma, Meg Tong, Tomek Korbak, David Duvenaud, Amanda Askell, Sam Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, et al. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, volume 2024, pages 110–144, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *International Conference on Machine Learning*, pages 111542–111558. PMLR, 2026.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In *EMNLP*. Association for Computational Linguistics, 2024.

Jyotika Singh, Fang Tu, Miguel Ballesteros, Weiyi Sun, Sandip Ghoshal, Michelle Yuan, Yassine Benajiba, Sujith Ravi, and Dan Roth. Mt-osc: Path for llms that get lost in multi-turn conversation. *arXiv preprint arXiv:2604.08782*, 2026.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Yuda Song, Lili Chen, Fahim Tajwar, Rémi Munos, Deepak Pathak, Drew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. In *International Conference on Machine Learning*, pages 114482–114522. PMLR, 2026.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Mudit Verma, Siddhant Bhambri, and Subbarao Kambhampati. On the brittle foundations of react prompting for agentic large language models. *arXiv preprint arXiv:2405.13966*, 2024.

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. In *International Conference on Machine Learning*, pages 67260–67283. PMLR, 2025a.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=VNckp7JEHn.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Berkeley function calling leaderboard. https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html, 2024.

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

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1028–1043, 2024.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3911–3921, 2018.

Hamed Zamani, Susan T. Dumais, Nick Craswell, Paul N. Bennett, and Gord Lueck. Generating clarifying questions for information retrieval. In *WWW*, 2020.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint arXiv:2501.09959*, 2025.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3713–3722, 2019.

Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5526–5543, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *International Conference on Machine Learning*, pages 162433–162448. PMLR, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning*, pages 62178–62209. PMLR, 2024.

George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Addison-Wesley Press, 1949.

Table 6: Inference-time interventions fail to recover the multi-turn gap. Each row applies a different lightweight intervention on top of the same baseline model in the SHARDED setting; accuracy (%) is reported on the same evaluation setup as Table 1.

Method	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
SHARDED (no intervention)	39.2	44.7	58.9	54.4	61.2
+ PROMPT-RULE	29.4	31.3	54.1	61.3	67.9
+ PROMPT-SELFCHECK	27.2	39.9	51.6	56.1	68.2
+ GATED-EXTERNAL (4o-mini)	28.5	36.8	45.6	37.9	42.5
+ GATED-SELF	30.7	4.4	37.7	30.0	63.0
+ FIC	**60.4**	**71.7**	**78.7**	**77.1**	**82.4**

A Inference-time Methods

We include four lightweight inference-time intervention baselines that do not modify model weights. Inference-time methods are a cheap alternative to post-training, and recent work shows that model behavior can be substantially improved without changing any parameters [Snell et al., 2025, Liu et al., 2025, Chen et al., 2025c]. The goal is to test whether the multi-turn gap can be closed with training-free methods. The interventions include (i) instructing the model to defer until enough information is present, and (ii) gating the model's answer attempts via a separate judge pass. See Appendix G for the exact prompts used.

Prompt-based interventions. PROMPT-RULE prepends a short behavioral rule to the system prompt instructing the model to defer until every numerical or factual value required to solve the task has been explicitly stated by the user. PROMPT-SELFCHECK requires the model to prefix every response with `READY: YES` or `READY: NO`, depending on whether it judges the available information sufficient. Only `YES` responses are scored; `NO` responses are treated as deferrals.

Judge-gated (two-pass) interventions. At every turn, an auxiliary judge first classifies whether the conversation so far contains enough information to attempt an answer. The assistant only generates an answer attempt when the judge returns `YES`. GATED-EXTERNAL uses GPT-4o-mini as the judge, while GATED-SELF uses a second pass of the assistant model itself with a judge-style system prompt.

Results. Table 6 reports SHARDED accuracy under each intervention. None of the four interventions consistently improves over the no-intervention SHARDED baseline; in many model–intervention combinations they actively degrade performance. These findings are consistent with prior observations that prompt-level interventions struggle to overturn ingrained model behavior [Huang et al., 2024, Sclar et al., 2024, Verma et al., 2024], and they motivate the parametric approach taken by FIC.

B Experimental Details

B.1 Compute Resource

All training and evaluation runs use $8\times$ A100 and H100 GPUs in a single node.

B.2 Evaluation Protocol

We follow the simulation protocol of Laban et al. [2026]. The user simulator is GPT-4o-mini at $T = 1.0$, with access to the full sharded instruction. At the first turn the user simulator always reveals the general goal (e.g., "How much ...?") and in subsequent turns it selects the next shard to reveal conditioned on the conversation history, mirroring how a real user would naturally introduce missing details. Assistant temperature is also fixed at $T = 1.0$. A strategy classifier categorizes each assistant response into one of seven types (clarification, refusal, hedging, interrogation, discussion, missing, or answer attempt); only answer attempts are scored. If an answer attempt is incorrect, the simulator continues revealing shards until a correct attempt is produced or the shard set is exhausted. We run $n = 10$ simulations per instance and report mean accuracy.

B.3 Multi-turn Corpus Construction

Source instructions and candidate filtering. Training shardings are produced from GSM8K [Cobbe et al., 2021] instructions disjoint from the 103-instance test split released by Laban et al. [2026]. We pre-filter to questions with at least three sentences, ensuring each problem can yield a non-trivial multi-shard decomposition.

Sharding pipeline. Each candidate is processed by the two-step "lazification" pipeline of Laban et al. [2026]: GPT-4o first segments the question into atomic facts, then rephrases each segment into a conversational shard $\{u_i\}$. The goal-revealing shard u_1 ("how many ...?") is always placed in the first turn; the remaining shards carry the supporting facts.

Verification filter. A candidate is admitted only if every base model in our experiment satisfies the information-equivalence criterion empirically: $\bar{P}(\pi; q) > 0$, $\bar{P}(\pi; q_{\text{concat}}) \geq \tau\, \bar{P}(\pi; q)$, and the same relation under a shuffled-concat presentation where the shards are randomly shuffled before concatenation. We restrict to shard counts $N \in [3, 8]$.

Intermediate-turn supervision (refusals). Intermediate assistant turns a_t ($t < N$) are synthesized by GPT-4o-mini, where roughly 45% of deferrals are phrased as questions and 55% as statements. Specific clarifying questions are retained only in the later turns, and *only* when a GPT-4o-mini judge verifies that the request actually targets information that will be revealed in the unrevealed shards.

Final-turn supervision. For each model and each problem we run $n = 10$ CONCAT rollouts at $T = 1.0$ and top-$p = 0.9$, and retain only the correct generations. One correct rollout is paired with each shard-order variant per epoch, so the final-turn target rotates across epochs (mirroring the stylistic variation applied to intermediate turns). When a problem has no correct CONCAT rollout for a particular backbone (a small minority of records), we fall back to the GSM8K reference solution. This makes the SFT signal *model-specific*: each backbone is supervised on its own solution traces, keeping the SFT distribution close to the model's prior.

B.4 Hyperparameters

LoRA targets are $\{q, k, v, o\}_{\text{proj}} \cup \{\text{gate}, \text{up}, \text{down}\}_{\text{proj}}$ for LLAMA-3.2-3B, LLAMA-3.1-8B, QWEN2.5-7B, and OLMO-2-13B, and the fused $\{qkv_{\text{proj}}, o_{\text{proj}}, \text{gate_up}_{\text{proj}}, \text{down}_{\text{proj}}\}$ for PHI-4-14B. We set LoRA rank r based on model size: $r = 8$ for LLAMA-3.2-3B, $r = 16$ for LLAMA-3.1-8B and QWEN2.5-7B, and $r = 32$ for OLMO-2-13B and PHI-4-14B. We hold $\alpha/r = 2$ and LoRA dropout 0.05 across all rank settings. SFT loss is the standard causal-LM cross-entropy computed only over assistant tokens. The learning rate for SFT is set to 2e-4 for all models. The effective batch size for SFT is 32 for smaller models (LLAMA-3.2-3B, LLAMA-3.1-8B, and QWEN2.5-7B) and 16 for larger ones (OLMO-2-13B and PHI-4-14B). The VASD LoRA adapter is initialized from the SFT checkpoint of the same backbone. We train with token-level Jensen–Shannon divergence loss for 1 epoch for each model. The learning rate for VASD is set to 1e-4 for all models. The effective batch size for VASD is the same as in the SFT stage.

Checkpoints are saved at the end of each epoch. SFT checkpoints are selected on a held-out development split, disjoint from both the training and the test split. VASD is a single-epoch pass with no checkpoint selection. We note that SFT checkpoint candidates perform similarly on the development split, so we additionally evaluate a fixed final SFT checkpoint for every backbone (Table 7). Every fixed-checkpoint value lies within the 95% confidence interval of the dev-selected one, the largest shifts are $+2.6$ and -1.5 points, and all conclusions are unchanged. We therefore recommend using the final SFT checkpoint directly.

B.5 Out-of-Domain Transfer Setup

Tasks. Database is built from Spider [Yu et al., 2018]: the assistant receives a database schema and a natural-language request and must produce an SQL query, which is scored against the reference. Relational databases underlie most enterprise data, and learning directly over them is an active direction on its own [Chen et al., 2025b]. Actions is built from the Berkeley Function Calling Leaderboard [Yan et al., 2024]: the assistant receives a tool specification and a user instruction

Table 7: FIC results with the dev-selected SFT checkpoint (as in Table 1) and with a fixed final SFT checkpoint, with 95% bootstrap confidence intervals. For LLAMA-3.1-8B and PHI-4-14B the dev-selected checkpoint is the final one, so the rows coincide.

Setting	Checkpoint	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
FULL	dev-selected	58.3 [51.1, 65.4]	73.8 [67.4, 79.8]	83.6 [78.2, 88.5]	79.5 [73.6, 85.2]	89.8 [85.0, 94.2]
	fixed	60.9 [53.9, 67.6]	73.8 [67.4, 79.8]	83.9 [78.7, 88.5]	80.5 [74.8, 85.6]	89.8 [85.0, 94.2]
CONCAT	dev-selected	52.6 [45.5, 59.6]	72.3 [65.8, 78.5]	82.6 [77.3, 87.5]	77.8 [71.6, 83.7]	88.0 [82.7, 92.6]
	fixed	54.0 [47.2, 60.7]	72.3 [65.8, 78.5]	85.2 [80.4, 89.6]	78.7 [72.8, 84.2]	88.0 [82.7, 92.6]
SHARDED	dev-selected	60.4 [53.6, 67.2]	71.7 [65.4, 77.8]	78.7 [73.1, 84.0]	77.1 [71.4, 82.5]	82.4 [77.2, 87.3]
	fixed	59.9 [53.4, 66.3]	71.7 [65.4, 77.8]	79.4 [74.0, 84.5]	75.6 [70.2, 80.8]	82.4 [77.2, 87.3]

Table 8: Main results of Table 1 with 95% hierarchical bootstrap confidence intervals over the 103 test instructions. Accuracy and single-turn attainment A in %.

Setting	Method	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
FULL	Baseline	62.5 [55.4, 69.4]	72.5 [65.9, 78.7]	84.2 [79.0, 88.9]	82.9 [77.5, 88.0]	88.3 [82.7, 93.2]
	FIC	58.3 [51.1, 65.4]	73.8 [67.4, 79.8]	83.6 [78.2, 88.5]	79.5 [73.6, 85.2]	89.8 [85.0, 94.2]
CONCAT	Baseline	60.2 [53.1, 67.2]	71.5 [64.7, 78.0]	85.0 [80.1, 89.6]	82.2 [76.4, 87.6]	89.4 [84.5, 93.9]
	FIC	52.6 [45.5, 59.6]	72.3 [65.8, 78.5]	82.6 [77.3, 87.5]	77.8 [71.6, 83.7]	88.0 [82.7, 92.6]
SHARDED	Baseline	39.2 [33.0, 45.8]	44.7 [38.3, 51.4]	58.9 [51.8, 65.9]	54.4 [47.6, 61.1]	61.2 [54.6, 67.8]
	FIC	**60.4** [53.6, 67.2]	**71.7** [65.4, 77.8]	**78.7** [73.1, 84.0]	**77.1** [71.4, 82.5]	**82.4** [77.2, 87.3]
Single-turn Attainment A		100 [92, 109]	100 [93, 108]	93 [87, 98]	94 [89, 99]	92 [87, 97]

and must emit a programmatic API call, which is checked for an exact abstract-syntax-tree match against the reference. Tool use underlies most LLM agents, and emitting correct API calls from natural-language instructions is an active area of study [Schick et al., 2023].

Adaptation. Starting from each model's FIC checkpoint trained on the Math task, we apply a lightweight final-turn-only SFT pass on the target domain. Concretely, for each domain we collect a small multi-turn corpus and supervise only the final-turn answer span. Intermediate-turn behavior is inherited from the math FIC checkpoint without further modification. No VASD pass is performed.

C Confidence Intervals for the Main Results

Table 8 repeats Table 1 with 95% confidence intervals from a hierarchical bootstrap: we resample the 103 test instructions with replacement and, within each resampled instruction, its 10 simulations. Contrasts and ratios use paired resampling on shared instructions, since all conditions are evaluated on the same instruction set. Three readings follow. First, the SHARDED gains are decisively outside noise: the baseline and FIC intervals are disjoint on every backbone, and the lower bound of the paired per-instruction difference is at least $+13.5$ points on all five. Second, both Llama backbones are statistically indistinguishable from full single-turn attainment, whereas the other three backbones measurably fall short, with upper bounds of 97–99%. The Llama intervals also reach down to 92–93%, so we do not claim that the two Llama backbones demonstrably differ from the other three. Third, for the single-turn FULL and CONCAT views, the intervals before and after training overlap on every backbone.

D Additional Controls

Non-cooperative User Stress Test. To probe behavior beyond the cooperative setting, we modify the user simulator of Laban et al. [2026] so that every user turn containing a value first presents a false value, phrased in varied ways, which the simulated user later retracts and corrects. The simulator is constrained so that the correction always appears after the initial error. Table 9 shows two examples. We evaluate the same FIC-trained LLAMA-3.1-8B checkpoint on 20 randomly selected test problems with 10 simulations per problem (200 conversations per condition). Accuracy is 69.5% $(139/200)$ under retractions versus 71.0% $(142/200)$ on the same problems in the cooperative setting.

Table 9: Examples of the retraction-and-correction perturbation applied to value-bearing user turns.

Original shard	Perturbed shard (false value)	Later correction
Lori plans to purchase a pair of shoes worth $320.00.	Lori plans to purchase a pair of shoes worth $250.00.	I had the shoe price wrong, they're $320.00
Their mother earns double the total of Valerie and her brother's salaries combined.	Their mother earns triple the total of Valerie and her brother's salaries combined.	Sorry, the mother earns double their combined total, not triple

Table 10: Over-deferral rate (%) on multi-turn SHARDED Math: the fraction of conversations with no answer attempt after all shards have been revealed.

Model	Baseline	FIC	Δ
LLAMA-3.2-3B	17.7	**3.1**	−14.6
LLAMA-3.1-8B	22.2	**3.3**	−18.9
QWEN2.5-7B	5.0	**0.9**	−4.1
OLMO-2-13B	12.4	**2.3**	−10.1
PHI-4-14B	13.1	**1.9**	−11.2

Table 11: Single-turn attainment (%) restricted to the test instances that the baseline CONCAT view solves, with 95% bootstrap confidence intervals.

Model	A [95% CI]
LLAMA-3.2-3B	98.1 [91.4, 105.3]
LLAMA-3.1-8B	98.2 [92.6, 104.3]
QWEN2.5-7B	92.1 [87.6, 96.6]
OLMO-2-13B	93.0 [89.1, 97.1]
PHI-4-14B	92.1 [87.9, 96.3]

Over-deferral. Under the evaluation protocol, the simulator stops once every shard has been revealed, and a response that is not an answer attempt at that point is scored as incorrect. Over-deferral is therefore penalized directly in accuracy, and the gains in Table 1 cannot come at its cost. Table 10 quantifies it as the fraction of conversations in which the assistant still produces no answer attempt after all information has been revealed: FIC reduces over-deferral by more than $5\times$ on every backbone, to a residual of 0.9–3.3%.

Attainment on teacher-solvable instances. Since the teacher runs on the CONCAT view, VASD could in principle propagate the teacher's own errors. Table 11 recomputes the single-turn attainment after dropping the test instances on which the baseline CONCAT view is incorrect in all 10 simulations. Attainment is essentially unchanged from the full-set values in Table 1.

Attainment against the post-training ceiling. The single-turn attainment in Table 1 divides by the baseline's CONCAT accuracy. Table 12 instead divides by each model's own CONCAT accuracy after FIC training, denoted A_{post}. LLAMA-3.1-8B stays at 99%, so its recovery is clean. LLAMA-3.2-3B exceeds 100% only because its CONCAT accuracy fell by 7.6 points after training, which marks a single-turn cost rather than surplus recovery (Section 4.2).

Self-generated SFT corpus. To quantify the contribution of external supervision, we rebuild the LLAMA-3.1-8B SFT corpus so that no external model or human-written text serves as a supervision target: all intermediate deferrals are generated by LLAMA-3.1-8B itself instead of GPT-4o-mini, all final-turn targets are the model's own correct CONCAT rollouts, and the 5% of instructions for SFT that would otherwise fall back to the GSM8K reference solution are excluded (the VASD corpus is fallback-free by construction). External LLMs are then confined to constructing the sharded user turns, which are verified information-equivalent task inputs and receive no loss. Evaluation is unchanged (103 instructions, 10 simulations each). Table 13 shows that after full FIC training the self-generated variant is within 1.6 points of the default corpus, with overlapping confidence intervals, and remains 25.4 points above the baseline. The external model is therefore an automation convenience in the data pipeline rather than a hidden teacher of the behavior. What the GPT-4o-mini pipeline adds is intermediate-turn helpfulness, namely requests for the specific missing detail, a conversational-quality property that final accuracy does not measure.

E Limitations and Future Work

We expand on the limitations noted in the main paper. First, the test splits released by Laban et al. [2026] contain on the order of 100 instructions per task, which limits the resolution of recovery-rate

Table 12: Single-turn attainment of FIC against the baseline CONCAT ceiling (A, as in Table 1) and against each model's own post-training CONCAT ceiling (A_{post}). *Above 100% because the CONCAT accuracy of LLAMA-3.2-3B drops after training.

	LLAMA-3.2-3B	LLAMA-3.1-8B	QWEN2.5-7B	OLMO-2-13B	PHI-4-14B
A (baseline ceiling)	100%	100%	93%	94%	92%
A_{post} (post-training ceiling)	115%*	99%	95%	99%	94%

Table 13: SHARDED accuracy (%) of LLAMA-3.1-8B with two sources of intermediate-turn SFT targets, with 95% bootstrap confidence intervals. In the self-generated variant no external model writes any supervision target.

Setting	SHARDED accuracy [95% CI]
Baseline (no training)	44.7 [38.3, 51.4]
FIC, GPT-4o-mini deferrals (default)	71.7 [65.4, 77.8]
FIC, self-generated deferrals	70.1 [63.9, 76.1]

estimates. We mitigate this by averaging $n = 10$ simulations per instance (amounting to roughly 1,000 scored conversations per model-condition cell), by reporting bootstrap confidence intervals (Appendix C), and by reporting consistent trends across five backbones and two out-of-domain tasks, but a larger, more diverse evaluation suite would strengthen the multi-turn-recovery claim. Second, the user simulator is cooperative and non-pivoting: it never changes sub-goals, retracts previously revealed information, or interleaves unrelated content. Our underspecification axis treats all shards as information-monotone (each shard adds, never removes, task-relevant information). Real users sometimes reveal contradictory or partially incorrect details that must be reconciled. Extending FIC to handle these richer conversational dynamics is a natural direction for future work. We view the present results as evidence that the simplest and most prevalent multi-turn failure mode—underspecification—is largely closeable with the model's own single-turn behavior as the only answer supervision. On the method side, VASD applies its dense distillation signal only at the final turn and leaves intermediate-turn deferral to SFT, which learns it cheaply. Supervising the intermediate turns by self-distillation as well, for instance with the same backbone conditioned on an explicit deferral instruction as the teacher, is a natural extension.

F Broader Impact

FIC improves the helpfulness and efficiency of LLMs in everyday underspecified conversations. The deferral behavior it instills—asking for missing information rather than hallucinating it—also has a positive honesty externality: users are less likely to receive confidently wrong answers based on assumed-but-unstated premises. The framework requires only modest compute and can be applied to any open-weight instruction-tuned model. We do not anticipate distinct misuse risks beyond those already present in the base backbones.

G System Prompts for Inference-time Interventions

The two prompt-based interventions modify only the assistant's system prompt; the user-simulator prompt, strategy classifier, answer extractor, and per-task evaluation prompts are all inherited unchanged from Laban et al. [2026]. The two judge-gated interventions (GATED-SELF and GATED-EXTERNAL) likewise reuse the Lost-in-Conversation default assistant system prompt for the answer pass; the auxiliary judge pass uses a short rubric prompt that asks whether the conversation so far contains sufficient information to attempt a final answer, returning `YES` or `NO`. The exact system prompts used for PROMPT-RULE and PROMPT-SELFCHECK on the math task are reproduced below.

PROMPT-RULE.

```
You are solving a mathematical problem where the user will provide
information gradually across multiple messages.
```

CRITICAL RULES:
1. Do NOT assume, fabricate, or guess any values not explicitly stated by the user. If a quantity is unknown, state that it is unknown -- never substitute a hypothetical value or use phrases like "let's assume" or "for example, $200 each".
2. Do NOT compute a final numerical answer until ALL quantities required for the calculation have been explicitly provided by the user. If any value is still missing, withhold computation and instead state what information is still needed.
3. When information is incomplete, respond with "I still need the following information to solve this: [list]" rather than filling in gaps yourself.
4. HOWEVER, as soon as you have enough explicitly stated values to perform the calculation, you MUST compute and present the final numerical answer immediately. Do not hesitate or ask for confirmation -- if the math can be done with the given values, do it.

PROMPT-SELFCHECK.

You are solving a mathematical problem where the user will provide information gradually across multiple messages.

Before EACH response, you must first determine if you have ALL the numerical values and information needed to compute the final answer.

Start every response with exactly one of:
READY: NO
READY: YES

If READY: NO -- State that information is still missing. Do NOT attempt any calculations and do NOT assume or fabricate any missing values.

If READY: YES -- Immediately compute the final answer using ONLY values explicitly stated by the user. Show your work and end with
FINAL ANSWER: <number>

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims reflect our contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 6 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full training and evaluation details are given in Appendix B, and code is available at `https://github.com/c-tl/FiC`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Data and code are available at https://github.com/c-tl/FiC.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.1 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 95% hierarchical bootstrap confidence intervals over test instructions for the main results and the key controls (Appendix C, Table 7, Table 11, Table 13).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix B.1 and the adaptation costs in Table 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper doesn't release new models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Mentioned and cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: External LLMs (GPT-4o and GPT-4o-mini) are used to construct the sharded multi-turn corpus, to phrase and verify intermediate-turn deferrals, and as the user simulator and strategy classifier in evaluation, following Laban et al. [2026]. Their roles are described in Section 3.1 and Appendix B. No external LLM provides final-answer supervision.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.