OnePred: Next-Query Prediction via Recursive Intent Memory in Multi-Turn Conversations

Anonymous ACL submission

Abstract

Although large language model (LLM) conversational systems process millions of multi-turn dialogues daily, they remain fundamentally reactive: they respond only after the user types a query. A key step toward proactive interaction is next-query prediction, which anticipates the user's subsequent query based solely on the preceding dialogue. Progress on this task is hindered by the lack of dedicated benchmarks and a fundamental efficiency–quality trade-off: naively concatenating full dialogue history incurs linearly growing token consumption, while truncating to the latest turn discards crucial cross-turn context. Our key insight is that accurate prediction does not require rereading raw history; it suffices to track the user's evolving intent trajectory across topics, unresolved needs, and interest shifts. We propose OnePred, which maintains a recursively updated memory as its sole cross-turn context, bounding the per-turn cost independently of conversation length. We train the model via a two-stage reinforcement learning pipeline that first teaches what to predict, then what to compress, shaping the memory into a prediction-oriented intent chain. To establish a rigorous testbed, we introduce NQP-Bench, spanning three diverse subsets. Experiments demonstrate that OnePred reduces per-turn token consumption by up to 22× compared to full-history inputs while consistently exceeding all baselines in prediction quality, with larger gains on longer conversations. Our code is anonymously available at https://anonymous.4open.science/r/OnePred-21F4.

1 Introduction

LLM-based conversational systems now process millions of multi-turn dialogues daily (Zheng et al., 2024), yet they remain fundamentally *reactive*: each response waits for the user to type. This passivity imposes real costs. Users must re-articulate needs that the system could have anticipated, retrieval pipelines cannot pre-fetch relevant documents, and latency accumulates over successive exchanges. We study *next-query prediction*, the task of forecasting a user's next query from the preceding dialogue, which enables the system to act before the user explicitly speaks.

Next-query prediction can shift conversational systems from reactive to proactive, enabling various downstream applications. The system can suggest follow-up questions to help users articulate unformulated needs, analogous to "People Also Ask" in web search. Predicted queries also enable speculative execution, allowing the system to pre-fetch documents, invoke retrieval-augmented generation pipelines, or compute candidate answers before the user submits, greatly reducing perceived latency. In addition, predicted queries allow routing infrastructure to dispatch anticipated requests to specialized models or knowledge bases ahead of time, improving throughput and response quality. More broadly, accurately predicting the next query suggests that the system has captured the user's evolving intent, a capability with direct implications for long-term user experience.

Despite this practical value, next-query prediction has received little dedicated study. Progress is impeded by both data scarcity and quality: authentic multi-turn logs are limited, and many real-world conversations are not naturally predictable, requiring careful curation to identify genuinely predictable instances. Related problems in adjacent fields do not fill this gap. *Query suggestion* in conversational systems typically optimizes for user clicks via preference alignment on implicit human feedback (Min et al., 2025; Yin et al., 2026), rather than modeling the natural trajectory of user intents. *Proactive dialogue* steers conversations toward specific goals (Deng et al., 2025, 2023; Wu et al., 2019) but focuses on system actions rather than anticipating user behavior.

Two straightforward approaches to handling dia-

logue history each fall short. A *Current-turn* model observes only the latest exchange. While efficient, it cannot detect topic continuations or recurring needs that span earlier turns, limiting its predictive capability in extended conversations. In contrast, a *Full-history* model concatenates all previous turns, providing rich context for prediction but at an increasing cost. As dialogue lengthens, the required context window grows linearly, demanding expensive long-context infrastructure or forcing lossy truncation. Moreover, predictive signals are buried under lengthy assistant responses and topical digressions, creating a signal-to-noise problem that simple concatenation cannot resolve. The core challenge is therefore one of compression: effective prediction requires cross-turn context, but only the subset that is informative for the user's next query.

Our key insight is that a model need not load the entire raw history into its context window. Instead, it only needs to track the user's intent chain: the evolving trajectory across topics, unresolved needs, and interest shifts. We represent this chain as a bounded free-form text state, allowing the LLM to read and write memory natively without additional modules while keeping the content directly interpretable. We propose **OnePred**, in which this memory is recursively updated at each turn. The model receives only the previous memory and the current user–assistant exchange, and decides what to retain, update, or discard. This design addresses both limitations above: it preserves cross-turn context that a single-turn window lacks, while bounding per-turn inference cost and filtering predictive signals from irrelevant context.

Learning to maintain such memory is non-trivial. Since there is no ground-truth annotation for what the memory should contain, its quality can only be evaluated through downstream prediction performance, making standard supervised fine-tuning (SFT) unsuitable. At the same time, optimizing memory poses a cold-start problem: effective compression requires knowing what signals matter for prediction, yet learning to predict through memory requires the memory to already carry useful context. To break this circular dependency, we adopt a two-stage reinforcement learning (RL) pipeline. **Stage 1** (Full-History RL) gives the model the complete conversation and trains next-query prediction directly. **Stage 2** (Agentic Memory RL) removes history access and trains the model to predict through its memory alone, forcing it to learn what to preserve. The ordering is essential: a model cannot learn to build useful memory without first knowing what information serves prediction.

To the best of our knowledge, no dedicated benchmark currently exists for open-ended next-query prediction in multi-turn LLM-assistant conversations, making it difficult to measure progress or compare approaches. To bridge this gap, we construct **NQP-Bench** (§2).

Our contributions are threefold:

- **Benchmark.** We construct **NQP-Bench**, to our knowledge the first dedicated benchmark for open-ended next-query prediction in multi-turn LLM-assistant conversations, spanning private, public, and cross-source settings, paired with a graded LLM-judge evaluation protocol validated against human annotators.
- **Method.** We propose OnePred, which maintains a recursive memory as a prediction-oriented intent chain. It is trained via a two-stage RL pipeline: Stage 1 learns to predict from full history, and Stage 2 learns to predict through bounded memory alone, thereby discovering what context should be preserved.
- **Empirical Findings.** OnePred reduces per-turn token consumption by up to 22× compared with full-history inputs while exceeding all baselines in prediction quality across datasets and model configurations, with its advantage increasing on longer conversations.

2 NQP-Bench

Given a multi-turn user–assistant conversation $\mathcal{C}_T = \{(q_t, r_t)\}_{t=1}^{T}$, the task of *next-query prediction* is to predict the user's next query q_{T+1} from the conversation context. To rigorously evaluate this capability, we require access to authentic multi-turn conversations where users interact naturally with LLM assistants. Consequently, we source data from real deployment logs and two large-scale public corpora, treating the last user query q_{T+1} in each session as the ground-truth prediction target and all preceding turns $\mathcal{C}_T$ as context. From these sessions we construct **NQP-Bench** (Next-Query Prediction Benchmark), comprising three subsets: **NQP-Priv**, drawn from a commercial conversational AI deployment and reflecting authentic user behavior; **NQP-Wild**, derived from WildChat (Zhao et al., 2024)[1] to support reproducibility; and **NQP-Share**, derived from ShareChat (Yan

[1] https://huggingface.co/datasets/allenai/WildChat-4.8M

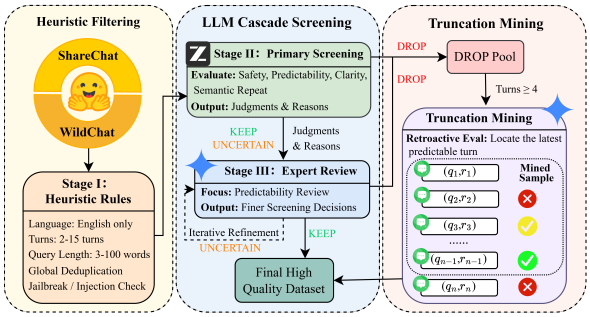


Figure 1: Overview of the NQP-Bench construction pipeline. The process integrates heuristic filtering, a two-stage LLM cascade for rigorous predictability screening, and a retroactive truncation mining strategy to salvage high-quality conversation prefixes from the DROP pool.

et al., 2025)[2] for cross-source generalization evaluation. NQP-Bench targets context-grounded next-query prediction rather than unconstrained future-behavior forecasting. To strictly protect user privacy, NQP-Priv will remain closed-source. The NQP-Wild and NQP-Share subsets are publicly released to support community research.

Dataset Construction. All three subsets undergo a unified multi-stage curation pipeline illustrated in Figure 1. Stage I applies heuristic rules to filter out low-quality noise. It enforces basic boundaries on turn counts and query lengths, removes degenerate patterns such as jailbreak attempts, and executes global deduplication for the public corpora.

Following heuristic filtering, an LLM cascade screening strategy spanning Stage II and Stage III is deployed to assess target predictability. Stage II serves as the primary screening phase using GLM-5 (Zeng et al., 2026) to evaluate safety, predictability, target clarity, and semantic repetition. The primary quality criterion is predictability: the core intent and key information of the target query must be inferable and learnable from the conversation history. To enforce this criterion, the model evaluates each sample against strict unpredictability rules, flagging cases such as random topic jumps or the sudden injection of user-private information. Based on this evaluation, each sample is categorized as KEEP, UNCERTAIN, or DROP, accompanied by a textual justification. Subsequently, Stage III employs a stronger expert language model, Gemini 2.5 Pro (Comanici et al., 2025), to review all KEEP and UNCERTAIN samples. The expert model reads the initial judgment and justification, evaluates whether the reasoning is valid, and decides whether to confirm or overturn the verdict. For boundary cases initially marked as UNCERTAIN, the expert model performs an iterative refinement process to resolve ambiguities and reach a final decision.

[2] https://huggingface.co/datasets/tucnguyen/ShareChat

Subset	Train	Test	Avg. Turns	Sources
NQP-Priv	21,707	2,212	4.79	Private
NQP-Wild	15,916	2,354	4.82	WildChat
NQP-Share	–	1,947	5.57	ShareChat

Table 1: Statistics of the NQP-Bench datasets. The benchmark provides a robust mix of private and public data for training, while reserving a dedicated cross-source subset exclusively for testing generalization.

To increase data yield from complex dialogues, we apply a truncation mining strategy to samples assigned to the DROP pool. For rejected conversations with four or more turns, we retroactively examine the dialogue history to identify the latest predictable turn. We then truncate the dialogue at this boundary, recovering high-quality predictable prefixes instead of discarding the entire session.

All retained samples are further annotated with three fine-grained labels (intent categories, cognitive difficulty, and intent transfer paradigms), with attrition rates and final statistics provided in Appendix D and Table 1. Averaging from 4.79 to 5.57 turns, NQP-Bench preserves interaction complexity, presenting a challenge for memory-driven reasoning.

Intent Evaluation Rubric. Evaluating generative predictions in open-ended conversations is challenging. Traditional string-matching metrics penalize intent-equivalent queries with different surface forms, while binary judgments fail to capture partial predictive success, such as predicting the correct topic but diverging in the specific question. To quantify partial contextual understanding, we introduce a 5-point Likert scale (Likert, 1932), shown in Table 2, based on our scoring protocol. This rubric accommodates the diversity of valid user intents and serves as the primary evaluation metric for both human and automated evaluators.

3 Method

OnePred consists of two components: a *recursive intent memory*, which is incrementally updated at each turn to maintain prediction-relevant context (§3.1), and a *two-stage RL training pipeline* designed to optimize both the model's predictive abil-

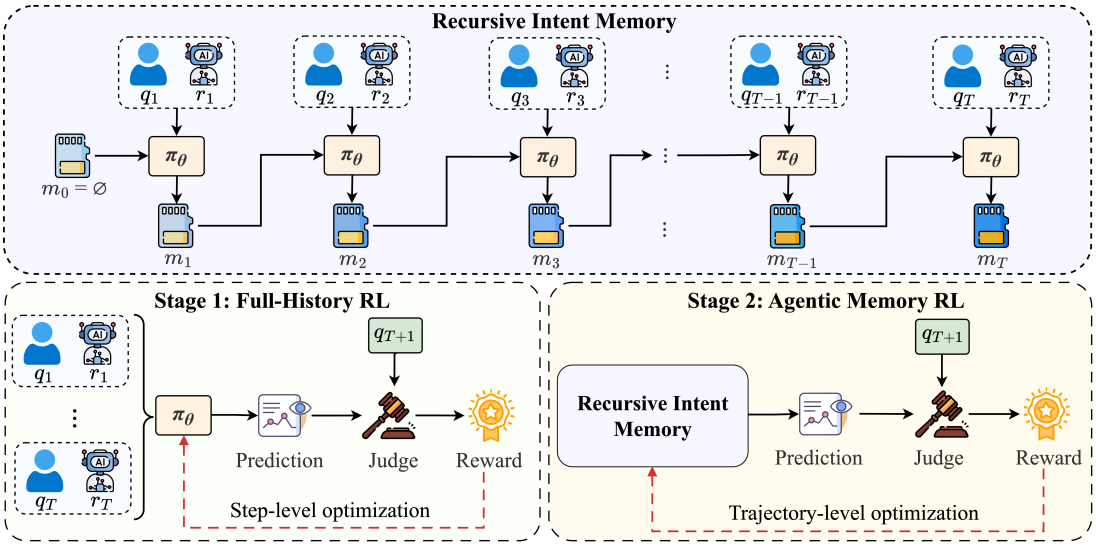


Figure 2: Overview of OnePred. **Top**: the recursive intent memory mechanism. At each turn, the model receives only the previous memory m_{t-1} and the current observation (q_t, r_t), and outputs an updated memory m_t. **Bottom**: the two-stage RL training pipeline. Stage 1 (Full-History RL) treats the entire conversation as a single-step input and directly optimizes prediction. Stage 2 (Agentic Memory RL) trains the model to predict through its memory alone over a multi-turn trajectory, broadcasting the final-turn reward to all preceding memory-update steps.

	Score Criterion
5	**Intent hit**: same core intent, equivalent phrasing
4	**Highly aligned**: same intent, different phrasing
3	**Topic related**: shared topic, different specific question
2	**Slightly related**: same domain, different direction
1	**Irrelevant**: no meaningful connection

Table 2: Intent evaluation rubric for scoring predicted queries against the ground truth.

ity and its memory compression strategy (§3.2). Figure 2 provides an overview.

3.1 Recursive Intent Memory

OnePred maintains a text-based memory state m_t that undergoes incremental updates as the conversation unfolds. At each turn, the model is conditioned exclusively on the previous memory state and the current user–assistant exchange, and decides which information to retain, update, or discard. Crucially, the raw conversation history is not accessible; the memory serves as the sole conduit for propagating historical context to the final prediction step. The memory is further bounded by a maximum length of k tokens, with any surplus content truncated before the next turn. This combination of memory-only context and token cap creates a hard information bottleneck, forcing the model to distill and retain only context that is useful for prediction.

Unlike a general-purpose summary that compresses information without a task-specific objective, OnePred's memory is shaped by the prediction reward (§3.2) to selectively retain prediction-relevant signals, forming what we term an **intent chain**. The intent chain is represented as free-form text: its content is not prescribed by a fixed schema, but emerges through RL optimization to encode the user's evolving topics and active needs.

Formally, given a T-turn conversation with observations $o_t = (q_t, r_t)$, the model takes the previous memory and the current observation as input, and generates an updated memory together with N candidate predictions:

$$m_t,\ \hat{q}_t^{(1)}, \ldots, \hat{q}_t^{(N)} = f_\theta(m_{t-1}, o_t), \qquad (1)$$

where $t = 1, \ldots, T$, the initial memory m_0 is empty, and f_θ denotes the language model, with no additional parametric module. Since the memory is text-based, the same LLM can both read and write it, while keeping its content directly interpretable (see Appendix E). The model outputs N candidates rather than a single prediction to account for the open-ended nature of multi-turn dialogue, where multiple plausible continuations may exist (see §3.2 for scoring).

This mechanism is used differently during inference and training. At inference time, the model emits both a memory update and candidate predictions after every turn, bounding the input size by at most $k + |o_t|$ tokens regardless of conversation length. This contrasts with Full-history inputs,

whose context grows linearly with T. During training, the model updates memory at each intermediate turn ($t = 1, \ldots, T-1$) and emits candidate predictions only at the final turn T. Intermediate memory updates receive no direct supervision, but are shaped indirectly through the final-turn reward (§3.2).

3.2 Two-Stage Training

The training objective is to optimize a single model f_θ that simultaneously maintains effective memory and produces accurate predictions. Supervised fine-tuning is not well suited to this task for two reasons. First, there is no ground-truth annotation specifying what the memory should contain. Second, next-query prediction is inherently open-ended, so requiring an exact match to a single reference would unfairly penalize semantically equivalent outputs. We therefore use reinforcement learning with an LLM-as-a-judge reward that assesses intent equivalence, allowing the final prediction reward to shape the memory content generated at earlier turns.

End-to-end memory training requires the model to learn both *what* is predictive and *how* to compress it into bounded memory. These two abilities are interdependent: effective compression requires knowing which signals matter for prediction, while learning to predict through memory requires the memory to already contain useful information. To break this circular dependency, we decompose training into two sequential stages, both optimized with Group Relative Policy Optimization (GRPO; Shao et al. 2024).

Stage 1: Learning to Predict (Full-History RL). We first train the model to predict q_{T+1} from the complete conversation history, bypassing the memory mechanism entirely. This stage allows the model to identify predictive signals, such as topic shifts and unresolved questions, without the additional difficulty of memory management. It thereby provides a strong initialization for the subsequent memory-learning stage.

Stage 2: Learning to Compress (Agentic Memory RL). Starting from the Stage 1 checkpoint, we switch to the memory interface defined in Eq. 1. The model now processes turns sequentially, conditioned exclusively on its previously generated memory and the current observation. Each rollout constitutes a T-step trajectory: for turns 1 through $T-1$, the model updates the memory state; at turn T, it outputs the final memory update together with N candidate predictions. The prediction objective and reward remain identical to Stage 1, but the full history is no longer accessible. The model is therefore forced to learn to preserve its predictive knowledge through bounded memory.

The two stages are complementary: Stage 1 establishes what to predict, while Stage 2 teaches how to compress the necessary context into bounded memory. Our ablation study (§4.3) confirms that neither stage alone matches the full pipeline.

Reward Design. Given N candidate predictions, the reward evaluates how well they capture the user's actual next query q_{T+1}. We adopt a best-of-N strategy: each candidate $\hat{q}^{(i)}$ is independently scored against the ground truth, and the highest score is taken as R_{judge}. The reward combines two components:

$$R = \lambda \cdot R_{\text{judge}} + (1 - \lambda) \cdot R_{\text{format}}. \tag{2}$$

R_{judge} is computed by an ensemble of LLM judges. Each judge independently rates the candidate–ground-truth pair on a discrete intent-alignment scale, and the final score is determined by majority vote (details in §4.1). This avoids relying on surface-level text-overlap metrics, such as BLEU and ROUGE, which cannot reliably capture intent equivalence between differently worded queries. R_{format} penalizes structural violations, such as missing delimiters or malformed outputs, ensuring that the agent loop remains well formed throughout training. The coefficient λ controls the trade-off between prediction quality and format compliance.

Credit Assignment. Both stages are optimized with the GRPO objective. For a given prompt x, we sample a group of G rollouts $\{y_1, \ldots, y_G\}$ from the sampling policy $\pi_{\theta_{\text{old}}}$. The importance ratio between the updated policy π_θ and the sampling policy is defined as:

$$\rho_i = \frac{\pi_\theta(y_i \mid x)}{\pi_{\theta_{\text{old}}}(y_i \mid x)}, \tag{3}$$

and the policy is optimized by minimizing

$$\mathcal{L}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \min\big(\rho_i \hat{A}_i,\ \bar{\rho}_i\, \hat{A}_i\big), \tag{4}$$

where $\bar{\rho}_i = \text{clip}(\rho_i,\ 1-\epsilon,\ 1+\epsilon)$. In Stage 1, x is the full conversation history; in Stage 2, x is the concatenation of the memory and the current observation at each turn. The advantage is normalized

within each group:

$$\hat{A}_i = \frac{R_i - \mathrm{mean}(\{R_j\}_{j=1}^{G})}{\mathrm{std}(\{R_j\}_{j=1}^{G})}. \tag{5}$$

In Stage 1, each rollout consists of a single prediction step, so $\hat{A}_i$ is directly applied. In Stage 2, each rollout is a multi-turn trajectory. Since the reward is only observed at the final prediction step T, yet heavily depends on the quality of earlier memory updates, we broadcast the trajectory-level advantage $\hat{A}_T$ uniformly to all preceding turns:

$$\hat{A}_t = \hat{A}_T, \quad t = 1, \ldots, T-1, \tag{6}$$

so that every memory-update step receives gradient signal from the eventual prediction outcome. This uniform broadcast is justified by the sequential dependency among memory states: each m_t causally influences all subsequent memories and the final prediction, making earlier updates jointly responsible for the final prediction quality.

4 Experiments

We evaluate our method along five dimensions: overall prediction quality (§4.2), the contribution of each training stage (§4.3), model scaling (§4.4), inference efficiency (§4.5), and robustness to dialogue length (§4.6). A qualitative case study is provided in Appendix E.

4.1 Experimental Setup

We evaluate on the three NQP-Bench subsets (§2) and compare three history interfaces: **Current-turn**, which uses only the most recent exchange; **Full-history**, which concatenates all preceding turns; and **Ours**, which uses recursive memory. Each interface is evaluated under three model regimes: **Gemini-3.1-Pro** (Google DeepMind, 2026), a frontier closed-source model; **Base Qwen**, Qwen3-8B (Qwen Team, 2025) without task-specific training; and **RL-trained Qwen**, Qwen3-8B optimized with two-stage RL. For the RL-trained regime, we train a separate checkpoint for each history interface on the NQP-Priv and NQP-Wild training splits. Specifically, the Current-turn and Full-history models are optimized using Stage 1 RL on their respective inputs, while our memory model follows the complete two-stage pipeline described in §3.2. NQP-Share is held out entirely during training, serving exclusively as an out-of-distribution (OOD) generalization test. The primary metric is the **LLM Judge** score, based on the rubric in Table 2 and mapped to $[0, 100]$. The final score is determined by majority vote among three commercial judges. We also report **Human** scores from expert annotators. Training details are provided in Appendix A.

4.2 Main Results

Table 3 presents the main comparison across three model regimes and three NQP-Bench subsets.

Consistent advantage across model regimes. Despite using only a bounded memory ($\leq k$ tokens) instead of the full conversation history, OnePred outperforms Full-history in all nine cells (three regimes $\times$ three datasets). Averaged across datasets, the Judge-score margin over Full-history is $+2.6$ for Gemini-3.1-Pro, $+1.9$ for Base Qwen, and $+1.6$ for RL-trained Qwen. The gap is largest for Gemini-3.1-Pro and smallest for RL-trained Qwen, which is expected because RL training benefits all history interfaces. For example, Full-history improves from 35.09 with Base Qwen to 42.43 with RL-trained Qwen, narrowing the absolute gap while preserving the overall ranking. The consistent gains across model capability levels, from an untrained 8B model to a frontier commercial API, suggest that the advantage stems from the memory-based history interface rather than model-specific artifacts. Human evaluation yields the same ranking in all nine comparisons, supporting the reliability of the LLM-judge protocol. We further compare against sliding-window and summarize-then-predict baselines in Appendix F, conduct hyperparameter sensitivity analyses in Appendix G, report bootstrap significance tests in Appendix H, analyze fine-grained performance in Appendix K, and provide rubric-level score distributions in Appendix J.

Private benchmark (NQP-Priv). On NQP-Priv, which reflects realistic deployment conditions with genuine user conversations, OnePred achieves the highest score in every model regime. The largest single margin appears under Gemini-3.1-Pro ($+3.4$ over Full-history), showing that recursive memory effectively captures cross-turn intent patterns in real user behavior. Notably, under Gemini-3.1-Pro, Current-turn and Full-history perform comparably (41.18 vs. 40.81), suggesting that naively concatenating noisy real-world dialogue history does not reliably improve prediction and may even slightly degrade performance. This observation further motivates a selective memory mechanism that retains

Method	NQP-Priv		NQP-Wild		NQP-Share		Average	
	Judge	Human	Judge	Human	Judge	Human	Judge	Human
Gemini-3.1-Pro								
Current-turn	41.18	42.10	45.53	45.31	42.78	43.33	43.16	43.58
Full-history	40.81	41.15	48.81	49.86	46.30	46.13	45.31	45.71
Ours	**44.22**	**44.85**	**50.72**	**51.14**	**48.77**	**49.66**	**47.90**	**48.55**
Base Qwen								
Current-turn	28.92	30.07	36.09	36.42	35.02	35.86	33.34	34.12
Full-history	31.37	31.85	38.20	39.16	35.69	36.30	35.09	35.77
Ours	**33.46**	**33.18**	**39.57**	**40.42**	**37.84**	**37.62**	**36.96**	**37.07**
RL-trained Qwen								
Current-turn	38.17	38.63	44.02	44.20	40.50	40.22	40.90	41.02
Full-history	40.79	40.64	44.35	44.97	42.16	42.53	42.43	42.71
Ours	**42.15**	**42.72**	**46.00**	**47.27**	**43.84**	**44.96**	**44.00**	**44.98**

Table 3: Main results on the three NQP-Bench subsets. For each subset, we report both the LLM-judge score (Judge) and the human evaluation score (Human). Rows are grouped by model regime: Gemini-3.1-Pro, Base Qwen, and RL-trained Qwen. The last two columns report metric-wise averages across the three subsets.

Method	Judge	Human
Base Qwen (no training)	39.57	40.42
+ Stage 1 only	43.82	44.63
+ Stage 2 only	42.96	43.35
+ Stage 1 then Stage 2	46.00	47.27

Table 4: Ablation of the two-stage training pipeline on NQP-Wild. All rows are evaluated with the recursive memory interface; only the training differs. Stage 1 only trains with full-history input but is evaluated under memory prompting without memory-specific RL.

Size	Ours (Base)		Ours (RL-trained)	
	Judge	Human	Judge	Human
1.7B	29.24	30.47	36.54	36.94
4B	34.07	33.88	42.34	42.09
8B	39.57	40.42	46.00	47.27

Table 5: Scaling results on NQP-Wild with 1.7B, 4B, and 8B Qwen3 backbones.

predictive signals while filtering out noise.

Public and cross-source benchmarks (NQP-Wild, NQP-Share). The same pattern holds on both public benchmarks. On NQP-Wild, our method leads Full-history by $+1.9$ (Gemini), $+1.4$ (Base), and $+1.7$ (RL-trained). Full-history benefits more from the longer, structured conversations in NQP-Wild than from noisy private logs (48.81 vs. 40.81 under Gemini), yet our method still outperforms it. On NQP-Share, which tests cross-source generalization, the advantage persists ($+2.5$ under Gemini and $+1.7$ under RL-trained). These results indicate that the learned memory interface transfers beyond the training distribution.

4.3 Ablation on the Two-Stage Training Pipeline

Table 4 ablates the contribution of each training stage on NQP-Wild.

Stage 1: learning to predict. Full-History RL alone brings a substantial improvement over the untrained Base Qwen ($39.57 \rightarrow 43.82$, $+4.3$). By training with complete conversation context, Stage 1 directly improves the model's next-query prediction ability and provides a strong initialization for subsequent memory training.

Stage 2: learning to compress. Agentic Memory RL from the Base model also yields meaningful gains ($39.57 \rightarrow 42.96$, $+3.4$), confirming that learning to predict through a compact memory is beneficial even without full-history pre-training.

Two-stage: complementary gains. The full two-stage pipeline achieves the best judge score of 46.00, exceeding Stage 1 alone by $+2.2$ and Stage 2 alone by $+3.0$. The gains are complementary: prediction skill from Stage 1 and compression skill from Stage 2 address different bottlenecks.

4.4 Scaling Analysis

Table 5 examines how OnePred scales with model size on NQP-Wild, using Qwen3 backbones with 1.7B, 4B, and 8B parameters. Without task-specific training, performance increases monotonically with scale: 29.24 (1.7B) $\rightarrow 34.07$ (4B) $\rightarrow 39.57$

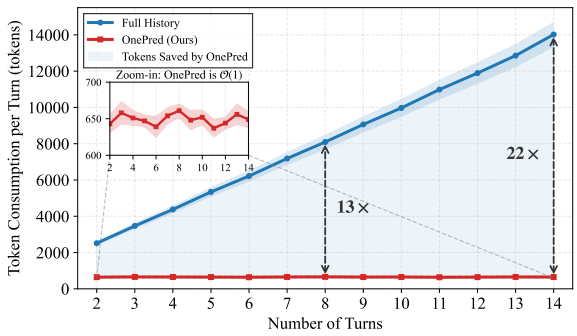


Figure 3: Average input tokens per turn for Full-history and OnePred across dialogue lengths on NQP-Wild. Full-history grows linearly with dialogue length, whereas OnePred remains bounded.

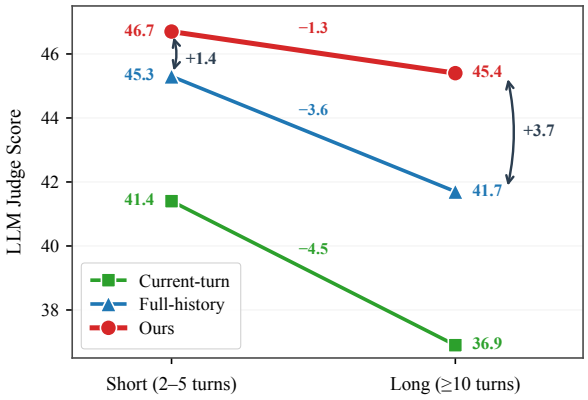


Figure 4: Performance by dialogue length on NQP-Wild with RL-trained Qwen. OnePred retains 97% performance on long dialogues (≥10 turns), while Full-history and Current-turn degrade more sharply.

(8B), indicating that larger capacity helps maintain coherent memory and predict intent. The two-stage RL pipeline yields consistent gains across all sizes: +7.3 (1.7B), +8.3 (4B), and +6.4 (8B). The largest absolute gain appears at 4B, while the 8B model already captures more predictive patterns from pre-training alone.

4.5 Inference Efficiency

OnePred substantially reduces inference cost. In a deployed system, next-query prediction runs after each turn, since the predictor must generate candidate queries before the user types. Full-history and OnePred therefore require the same number of LLM calls (one per turn); the difference lies in the input size of each call. For Full-history, the prompt at turn t includes all t previous exchanges, causing the context length to grow linearly with conversation length. In contrast, OnePred conditions on the system prompt, bounded memory ($\leq k$ tokens), and the current observation, making the input size independent of the number of prior turns.

Figure 3 quantifies this difference on NQP-Wild. Our method uses roughly 650 tokens per turn regardless of conversation length. Full-history starts at ∼2,500 tokens for a 2-turn dialogue and grows to over 14,000 tokens by turn 14, reaching a 13× gap at turn 8 and 22× at turn 14. This gap persists even with KV caching, since each generated token still attends to all cached states, keeping decode-time cost proportional to sequence length. By bounding the input, OnePred ensures consistent decode latency regardless of conversation length.

4.6 Performance by Dialogue Length

To analyze how each method handles longer conversations, we split NQP-Wild into short conversations with 2–5 turns and long conversations with at least 10 turns, and report the Judge score of RL-trained Qwen for each group in Figure 4.

All methods perform worse on long conversations, but the degree of degradation differs substantially. Current-turn drops by 4.5 points (41.4 → 36.9), as a single exchange provides increasingly insufficient context. Full-history drops by 3.6 points (45.3 → 41.7), likely because longer prompts introduce more irrelevant context and dilute predictive signals. In contrast, OnePred drops by only 1.3 points (46.7 → 45.4), retaining 97% of its short-conversation performance. Its advantage over Full-history widens from +1.4 points on short conversations to +3.7 points on long conversations, indicating that recursive memory is particularly effective as conversations grow longer.

5 Conclusion

We formalize next-query prediction in multi-turn LLM conversations and introduce OnePred, which maintains a recursively updated intent memory for proactive prediction without full-history concatenation. Experiments on NQP-Bench show that OnePred reduces per-turn input tokens by up to 22× compared with Full-history while outperforming all baselines in prediction quality. The bounded memory acts as a task-oriented information bottleneck that filters conversational noise, yielding robust performance as conversations lengthen. We release the public subsets of NQP-Bench and OnePred as a scalable, interpretable step from reactive response generation toward proactive interaction.

Limitations

The bounded memory keeps per-turn cost constant but inevitably loses some fine-grained details during compression, such as specific numbers, exact phrasing, or minor sub-topics mentioned in passing. Appendix K shows cases where such details become relevant to the next query. Our RL training and open-source experiments use Qwen3 backbones ranging from 1.7B to 8B. Although we also evaluate Gemini-3.1-Pro, we have not tested other backbone families such as Llama or Mistral. Finally, LLMs are involved in both benchmark curation and evaluation scoring. Although we use architecturally distinct models for these two stages and validate curation quality with human annotators ($\kappa = 0.83$; Appendix I), some residual shared bias between LLM curation and LLM evaluation cannot be fully ruled out. Human evaluation yields the same method ranking across all comparisons, but the two protocols may still disagree on individual samples.

Ethical Considerations

Our work uses two public datasets under their respective terms: WildChat (Zhao et al., 2024), which was collected with users' informed consent and released after PII removal, and ShareChat (Yan et al., 2025), which was collected from user-shared conversation URLs under IRB approval and de-identified using Microsoft Presidio with auxiliary model verification. Our curation pipeline only selects predictable conversation prefixes without altering the original content. The private NQP-Priv subset is derived from internal deployment logs and is not released; all results are reported in aggregate, with no individual conversations or user identifiers exposed. The use of these logs complies with the platform's terms of service, under which users consented to anonymized data being used for service improvement and research. All conversations were de-identified prior to annotation, and the study was reviewed and approved under the organization's internal data governance policy. Human annotators were informed of the task purpose and compensated at local market rates. Annotators were recruited from graduate students or professional annotators with experience in dialogue evaluation, and no personally identifying information about annotators is released. Because next-query prediction models user intent trajectories and may be misused for profiling or manipulation, deployment should be limited to assistive settings with clear user consent, opt-out mechanisms, strict data governance, and safeguards against using predicted intents for advertising, behavioral profiling, or other non-assistive purposes.

References

Anthropic. 2026. Claude opus 4.7 system card. https://www.anthropic.com/system-cards. Model system card for Claude Opus 4.7.

Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query recommendation using query logs in search engines. In *International conference on extending database technology*, pages 588–596. Springer.

Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883.

Jinwen Chen, Shuai Gong, Shiwen Zhang, Zheng Zhang, Yachao Zhao, Lingxiang Wang, Haibo Zhou, Yuan Zhan, Wei Lin, and Hainan Zhang. 2026. Localsug: Geography-aware llm for query suggestion in local-life services. *arXiv preprint arXiv:2603.04946*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, and Evan Rosen. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*.

Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. 2025. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45.

Google DeepMind. 2026. Gemini 3.1 pro model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf. Accessed: 2026-05.

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

Xian Guo, Ben Chen, Siyuan Wang, Ying Yang, Mingyue Cheng, Chenyi Lei, Yuqing Ding, and Han Li. 2026. Onesug: The unified end-to-end generative framework for e-commerce query suggestion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40(17), pages 14774–14782.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Erxue Min, Hsiu-Yuan Huang, Xihong Yang, Min Yang, Xin Jia, Yunfang Wu, Hengyi Cai, Junfeng Wang, Shuaiqiang Wang, and Dawei Yin. 2025. Ctr-guided generative query suggestion in conversational search. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2624–2634.

OpenAI. 2026. Gpt-5.5 system card. https://openai.com/index/gpt-5-5-system-card/. Accessed: 2026-05.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Penghui Qi, Runqi Lin, Junliang Guo, and Wei Xu. 2024. Verl: An extensible framework for post-training of large language models. *arXiv preprint arXiv:2409.19256*.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562.

Jinqiang Wang, Huansheng Ning, Jianguo Ding, Tao Zhu, Liming Chen, and Chris Nugent. 2025. Llm-driven preference data synthesis for proactive prediction of the next user utterance in human-machine dialogue. *arXiv preprint arXiv:2601.09713*.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804.

Yueru Yan, Tuc Nguyen, Bo Su, Melissa Lieffers, and Thai Le. 2025. Sharechat: A dataset of chatbot conversations in the wild. *arXiv preprint arXiv:2512.17843*.

Junhao Yin, Haolin Wang, Peng Bao, Ju Xu, and Yongliang Wang. 2026. From clicks to preference: A multi-stage alignment framework for generative query suggestion in conversational system. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2539–2550.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, and Mingxuan Wang. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, and Jie Tang. 2026. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntao Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

A Implementation Details

Training setup. The backbone model is Qwen3-8B, trained with the verl framework (Sheng et al., 2024) for GRPO. Both stages use the AdamW optimizer with a PPO clip ratio $\epsilon = 0.2$ (Eq. 4). **Stage 1** (Full-History RL): 500 steps, constant learning rate 1×10^{-6}, batch size 32, 8 rollouts per prompt, maximum prompt length 8,192, maximum response length 3,072. **Stage 2** (Agentic Memory RL): initialized from the Stage 1 checkpoint, 200 steps, learning rate 5×10^{-7} with cosine schedule and 2% warmup, batch size 32, 8 rollouts per prompt, maximum response length 4,096. The increased response length in Stage 2 accommodates the multi-turn trajectory format (memory update at each intermediate turn, with prediction appended only at the final turn). The reward follows Eq. 2 with $\lambda = 0.9$; the memory budget is $k = 500$ tokens per turn; the number of candidate predictions is $N = 3$. Sensitivity analyses for k, N, and λ are provided in Appendix G. For the scaling experiments (Table 5), the 1.7B and 4B models are trained with the same hyperparameters as the 8B model.

Compute budget. Stage 1 and Stage 2 training were conducted on 8 NVIDIA A100 80GB GPUs, taking approximately 256 GPU-hours in total for the 8B model (including both stages and all rollout generation).

Inference. During evaluation, all models decode greedily (temperature 0). The memory budget is enforced by hard truncation at k tokens after each memory update; if the model's output exceeds k tokens in the memory field, the suffix beyond k tokens is discarded before the next turn.

Evaluation details. The LLM Judge score is determined by majority vote of three frontier models: Claude Opus 4.7 (Anthropic, 2026), GPT-5.5 (OpenAI, 2026), and Gemini-3.1-Pro (Google DeepMind, 2026). Each judge independently rates every candidate–ground-truth pair on the 5-point rubric (Table 2), and the majority rating is taken as the final score (median when no majority exists), linearly mapped to $[0, 100]$ via $(s-1)/4 \times 100$ where $s \in \{1, 2, 3, 4, 5\}$. The same ensemble is used for the training reward R_{judge}. For human evaluation, five expert annotators independently scored 250 randomly sampled instances per subset on the same rubric, and their majority vote is taken as the final human score. Further validation of the judge protocol is provided in Appendix I.

B Related Work

B.1 Query Suggestion and Proactive Dialogue

Query suggestion (QS) has evolved from traditional search environments (Baeza-Yates et al., 2004; Cao et al., 2008; Sordoni et al., 2015; Dehghani et al., 2017) to conversational AI. Recent generative frameworks extend QS to proactive conversational settings, but are often designed for constrained domains, such as e-commerce (Guo et al., 2026), local services (Chen et al., 2026), and click-through-rate maximization (Min et al., 2025; Yin et al., 2026). Importantly, these QS systems typically formulate the task as preference alignment over implicit feedback logs to induce user clicks. This differs from our next-query prediction task, which aims to model the natural trajectory of user intent.

Proactive dialogue systems typically steer conversations toward system-defined goals, such as task completion (Deng et al., 2023; Wu et al., 2019). Rather than anticipating the user's autonomous intent, they focus on selecting the next optimal system action. Among existing approaches, Wang et al. (2025) is most closely related to our work, as it anticipates user queries through offline-synthesized intent trees. However, their approach relies on heuristic structures and still requires full dialogue history at inference time, leaving the computational bottleneck unresolved. In contrast, OnePred tracks dynamic user intents through a bounded, recursively updated memory chain, avoiding both static structures and full-history concatenation.

B.2 Memory-Augmented LLMs and Reinforcement Learning

Managing long-range context in multi-turn dialogues commonly relies on full-history concatenation or retrieval-augmented generation. Although effective in preserving context, these paradigms suffer from increasing inference costs and signal dilution (Liu et al., 2024; Wang et al., 2023). Recent work introduces RL to maintain bounded memory for factual retrieval and task execution across extended contexts (Yu et al., 2025; Zhou et al., 2025). While OnePred shares the principle of RL-driven selective retention, it optimizes a different target: rather than preserving factual information, it learns a forward-looking memory that retains evolving intent trajectories for next-query prediction.

Training such a memory further introduces a credit-assignment challenge. Standard RL appli-

cations, including GRPO (Shao et al., 2024; Guo et al., 2025), are typically applied to single-step generation. In contrast, OnePred optimizes a sequence of memory states without intermediate supervision, where the final prediction depends on all preceding memory updates. To address this, OnePred broadcasts the final-turn trajectory advantage across preceding memory-update steps, adapting GRPO to multi-turn conversational prediction.

C Dataset Statistics and Characteristics

This section presents the statistical characteristics of NQP-Bench. All structured metadata used in this analysis were automatically annotated using Gemini 3.1 Pro. Through visual analysis of intent distributions, dialogue lengths, intent transfer dynamics, and difficulty, we demonstrate that NQP-Bench is a highly diverse and challenging benchmark suitable for evaluating next-query prediction in multi-turn interactions.

C.1 Intent Diversity

NQP-Bench employs a fine-grained intent taxonomy consisting of 7 primary and 17 secondary intents. Figure 5 illustrates the intent distribution across the WildChat and ShareChat subsets. The data spans a wide spectrum of user needs, anchored by three major domains: *Knowledge QA* (focusing on factual queries and domain-specific interpretation), *Coding* (spanning code generation and debugging), and *Creative Writing*. This comprehensive distribution ensures that models are evaluated on their predictive capabilities across diverse real-world application scenarios, rather than overfitting to a single domain.

C.2 Conversation Depth

To evaluate a model's ability to track long-term intent trajectories, a multi-turn benchmark must preserve sufficient conversational depth. As shown in Figure 6, both NQP-Wild and NQP-Share exhibit a healthy distribution of conversation lengths. Notably, with averages of 4.82 (NQP-Wild) and 5.57 (NQP-Share) turns, the datasets maintain a substantial "long-tail" of deep conversations (≥ 10 turns). This long-tail volume provides the crucial data foundation for the long-context performance analysis discussed in §4.6.

C.3 Dynamics of Intent Transfer

Users in multi-turn dialogues rarely follow a static path. We categorize the dynamic evolution of user intent into four paradigms, defined by their core behavioral characteristics:

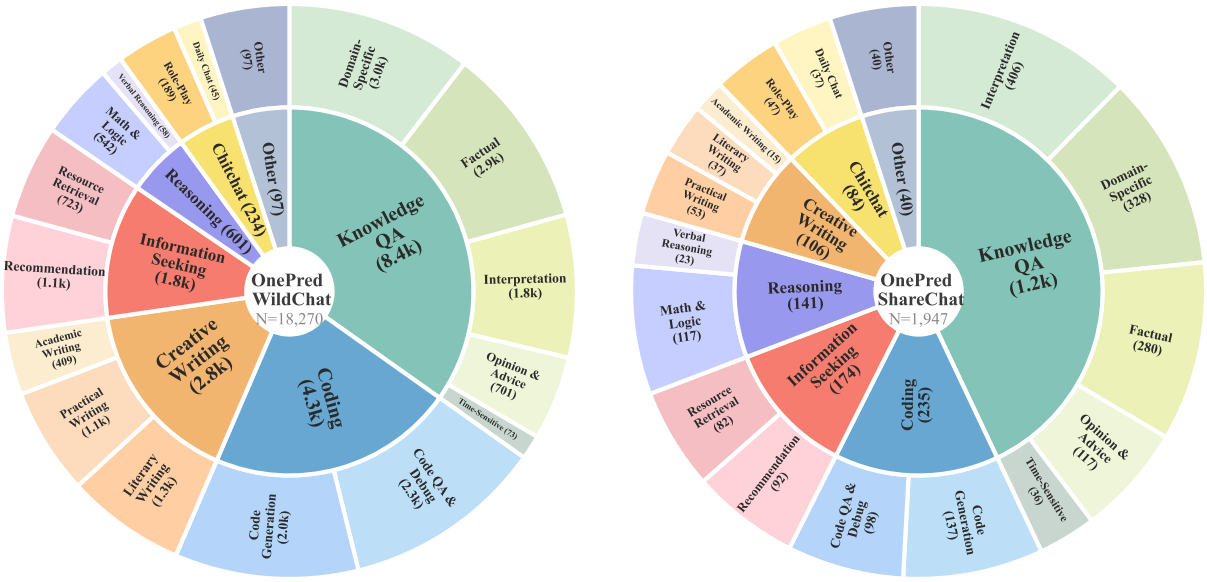
- **Deepening:** Focuses on conceptual comprehension, where the user stays on the same core topic but goes deeper to seek a more sophisticated understanding.
- **Application:** Operates at the practical execution level, where the user performs further operations on the AI's previous output to produce a derived artifact.
- **Associated Shift:** Represents logically inferable topic transitions, where the user switches to a different but contextually connected topic.
- **Challenge:** Centers on contradiction and conflict resolution, where the user pushes back on the AI's output to correct substantive errors or flawed assumptions.

Figure 7 reveals that while *Deepening* accounts for over half of the transitions (aligning with the natural human tendency to drill down into a topic), nearly 40% of the queries involve *Application* or *Associated Shift*. This confirms that accurate next-query prediction requires the model to anticipate dynamic intent evolutions rather than simply extending the current context.

C.4 Difficulty

To ensure the benchmark possesses sufficient headroom to distinguish advanced models, we classify the cognitive difficulty of each prediction target using a three-dimensional gating framework. A prediction is classified as **Hard** if it triggers any of the following conditions:

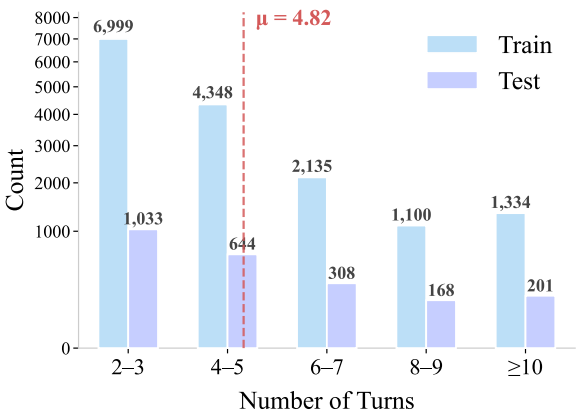
- **Contextual Distance:** The prediction requires information from ≥ 2 turns ago (i.e., it cannot be predicted using only the latest exchange); for *Associated Shift*, the polarity is reversed because a distant anchor makes the shift more predictable, whereas a jump without prior grounding is inherently harder.
- **Predictive Entropy:** The current dialogue state presents ≥ 5 independent and equiprobable future directions.
- **Reasoning Gap:** Establishing the connection between the dialogue history and the target query requires implicit reasoning or external domain knowledge.



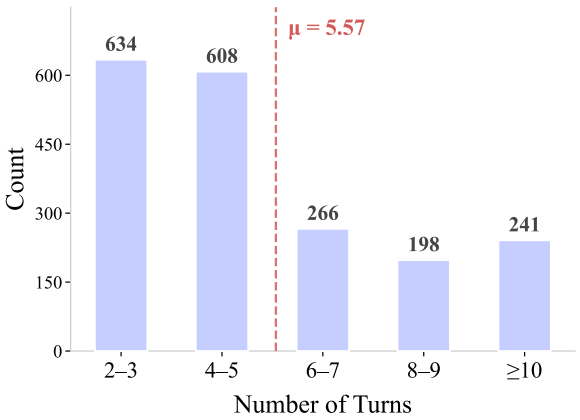
(a) Intent Distribution (NQP-Wild)

(b) Intent Distribution (NQP-Share)

Figure 5: Sunburst charts detailing the hierarchical intent taxonomy and their distributions. The datasets demonstrate high diversity across knowledge-seeking, reasoning, and generative tasks.



(a) Turn Distribution (NQP-Wild)



(b) Turn Distribution (NQP-Share)

Figure 6: Distribution of conversation lengths. The benchmark retains a significant proportion of long conversations ($\geq$ 10 turns), establishing a robust testbed for evaluating memory and cross-turn intent tracking.

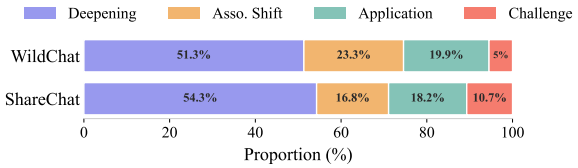


Figure 7: Proportion of intent transfer paradigms. Over 40% of the transitions involve cognitive leaps (Application or Associated Shift) rather than simple topic continuation, highlighting the dynamic nature of user intents.

As depicted in Figure 8, under this rigorous non-linear criterion, 41.6% of NQP-Wild and 52.0% of NQP-Share samples are categorized as *Hard*. This substantial proportion of challenging instances prevents models from achieving high scores via superficial pattern matching, ensuring a rigorous evaluation of true agentic reasoning.

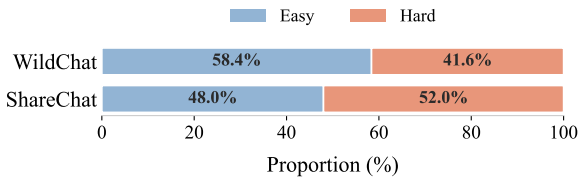


Figure 8: Distribution of sample difficulty. Nearly half of the benchmark consists of *Hard* instances, requiring long-context dependency, disambiguation of high-entropy states, or implicit reasoning.

D Curation Pipeline Attrition

Table 6 reports the number of samples retained at each pipeline stage for the two public subsets.

Stage	NQP-Wild		NQP-Share	
	#Samples	Ret.%	#Samples	Ret.%
Raw corpus	3,199K	–	130K	–
Stage I (heuristic)	148K	4.6	30.5K	23.6
Stage II (GLM-5)	29.2K	19.7	6.1K	20.0
Stage III (Gemini)	21.0K	72.0	4.7K	77.2
+ Trunc. mining	+6.7K	–	+1.2K	–
Final	18.3K	–	1.9K	–

Table 6: Filtering attrition for the two public NQP-Bench subsets. "Ret.%" is the retention rate relative to the previous stage. Truncation mining recovers predictable prefixes from the Stage II DROP pool.

Stage I attrition in WildChat (95.4%) is dominated by language filtering (the corpus is multilingual; only English is retained). Stage II applies the most aggressive filter: $\sim$80% of heuristically clean samples are judged unpredictable, reflecting the inherent difficulty of the task. Stage III confirms 72–77% of forwarded samples, overturning roughly one quarter upon expert review. After pooling Stage III KEEP and truncation-mined samples, a final quality pass removes near-duplicates (by embedding similarity) and over-length samples, reducing the pool to the released benchmark size, followed by a stratified train/test split balance of difficulty (Appendix C).

E Case Study

Figure 9 presents a representative 13-turn example from NQP-Wild illustrating how recursive memory supports prediction across extended, topic-shifting conversations. The user begins with questions about protein content in foods (nutrition), transitions through health conditions, chlorogenic-acid-rich foods, cooking recipes, and finally arrives at skin collagen, spanning four distinct topic clusters over 13 turns. The ground-truth next query is "Which foods are best to increase collagen production?", a question that bridges the user's early nutritional interest (turns 1–11) with the recent collagen focus (turns 12–13).

This cross-topic synthesis exposes the limitations of both baselines. A **Current-turn** model sees only the final exchange about skincare routines for collagen retention; without access to the earlier food-and-nutrition thread, it lacks the signal needed to predict a food-oriented follow-up and would instead predict skincare-related questions. A **Full-history** model receives all 13 turns (>14K tokens), but the predictive signal, the user's persistent interest in food and nutrition, is buried among recipe corrections, a digression on numbness symptoms (turns 3–5), and lengthy assistant responses, making it harder to distill the relevant cross-topic thread.

Our method compresses 17.6K characters of raw conversation into a 338-character memory (52× compression) that selectively retains the evolving intent trajectory: *food/nutrition* → *health* → *cooking* → *skin-health/collagen*. From this compact state, the model correctly predicts "What foods are best for increasing collagen production?" as its top candidate, achieving an exact intent match with the ground truth. The example illustrates the core advantage of recursive memory: it discards turn-level noise while preserving the cross-topic predictive signal that neither a single-turn window nor an uncompressed full history can reliably surface.

F Additional Baselines

To isolate the effect of history interface design from task-specific training, we compare all five interfaces using Gemini-3.1-Pro in a zero-shot setting on NQP-Wild, ensuring no method benefits from dedicated optimization.

- **Sliding-window** (w=3): The model receives only the most recent $w = 3$ user–assistant exchanges as input, discarding all earlier turns.
- **Summarize-then-predict**: At each turn, the model first generates a free-form summary of the full conversation history in a single pass, then predicts the next query conditioned on this summary.

Method	Judge	Human
Current-turn	45.53	45.31
Sliding-window (w=3)	46.15	46.38
Summarize-then-predict	47.42	47.80
Full-history	48.81	49.86
Ours	**50.72**	**51.14**

Table 7: Comparison with additional baselines on NQP-Wild (Gemini-3.1-Pro, zero-shot). All methods use the same model without task-specific training, isolating the effect of history interface design.

As shown in Table 7, the ranking Ours > Full-history > Summarize-then-predict > Sliding-

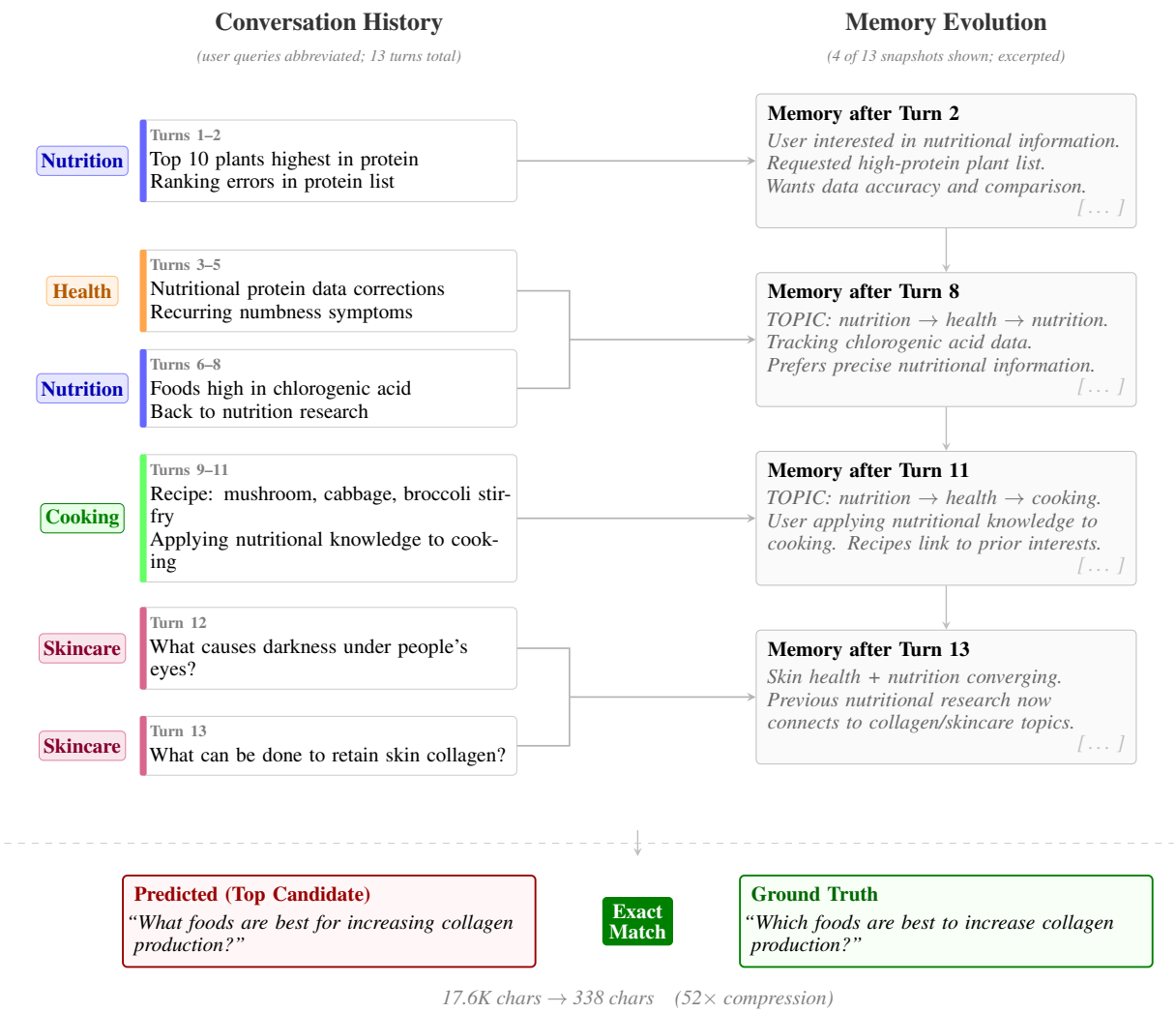


Figure 9: Qualitative example from NQP-Wild. Left: abbreviated user queries from a 13-turn conversation spanning four topic clusters (assistant responses omitted). Right: four representative snapshots of the recursive memory (excerpted; the full memory is updated at every turn). The memory compresses 17.6K characters into 338 characters (52×) while preserving the cross-topic intent trajectory, enabling an exact intent match with the ground-truth next query.

window > Current-turn holds consistently. Sliding-window (w=3) provides only a marginal gain over Current-turn (+0.62), confirming that a fixed recent window captures little additional predictive signal beyond the latest exchange.

Summarize-then-predict outperforms Current-turn (+1.89), showing that a frontier model can extract useful cross-turn context through summarization. However, it still falls short of Full-history (−1.39), indicating that even high-quality task-agnostic compression loses predictive signals that raw history preserves. Our recursive memory outperforms Full-history by +1.91 and Summarize-then-predict by +3.30, demonstrating that the advantage stems from the structure of prediction-oriented recursive compression, not merely from having access to cross-turn context.

G Hyperparameter Sensitivity

We ablate inference-time hyperparameters (k and N) using Gemini-3.1-Pro in a zero-shot setting on NQP-Wild, isolating the effect of each parameter from task-specific training. We separately ablate the training-time reward weight λ using RL-trained Qwen, where each value is independently trained.

Memory budget k. Performance increases steadily from k=100 to k=500 as the model gains capacity to retain richer intent trajectories. Beyond k=500, performance plateaus and slightly declines: with a larger budget, the memory tends to retain verbose turn-level details that dilute predictive sig-

k (tokens)	Judge	Human	Avg. Tokens/Turn
100	47.80	48.20	∼250
200	49.35	49.70	∼350
500	**50.72**	**51.14**	∼650
800	50.50	50.85	∼950
1,000	50.25	50.60	∼1,150

Table 8: Effect of memory budget k on NQP-Wild (Gemini-3.1-Pro, zero-shot).

nals. The sweet spot at k=500 balances sufficient capacity with effective information bottlenecking. Even at k=1,000 the per-turn cost (∼1,150 tokens) remains far below Full-history (∼14,000 tokens at 14 turns).

N	Judge	Human
1	49.15	49.80
2	50.22	50.70
3	**50.72**	**51.14**
5	50.85	51.28

Table 9: Effect of the number of candidate predictions N on NQP-Wild (Gemini-3.1-Pro, zero-shot).

Number of candidate predictions N. N=1 incurs a notable penalty (−1.57), as a single prediction must cover all ambiguity in the user's next intent. The gain from N=2 to N=3 is moderate (+0.50), and N=5 offers only marginal further improvement (+0.13), indicating that three candidates suffice to cover the main intent hypotheses without wasting generation budget.

Reward weight λ. Since λ is a training-time parameter, each value below corresponds to an independently trained model (RL-trained Qwen on NQP-Wild).

λ	Judge	Human
0.5	44.85	45.60
0.7	45.48	46.30
0.9	**46.00**	**47.27**
1.0	45.62	46.55

Table 10: Effect of reward weight λ on NQP-Wild (RL-trained Qwen).

When λ is too low (0.5), the format penalty dominates and the model under-optimizes for prediction quality. At λ=1.0 (pure judge reward, no format penalty), performance drops slightly because occasional malformed outputs corrupt the multi-turn agent loop during training. λ=0.9 strikes the best balance, allocating most of the reward signal to prediction quality while maintaining sufficient format compliance to keep training stable.

H Statistical Significance

To verify that our improvements are statistically reliable, we report bootstrap confidence intervals for the main comparison on NQP-Wild (RL-trained Qwen). We resample the test set (n=2,354) 1,000 times with replacement and compute the LLM Judge score for each bootstrap sample. We further compute paired bootstrap p-values by counting the fraction of resamples in which the baseline score exceeds ours.

Method	Judge (mean ± 95% CI)	p-value vs. Ours
Current-turn	44.02 ± 0.78	< 0.001
Full-history	44.35 ± 0.82	< 0.01
Ours	46.00 ± 0.75	—

Table 11: Bootstrap confidence intervals and paired bootstrap p-values on NQP-Wild (RL-trained Qwen, 1,000 resamples).

The 95% confidence intervals of our method do not overlap with those of either baseline, and both paired p-values are well below 0.01, confirming that the improvements are statistically significant.

I Judge Validation

To verify that the LLM-judge protocol is a reliable proxy for human judgment, we analyze agreement at three levels using the 250 human-annotated samples per subset.

Human inter-annotator agreement. Five expert annotators independently scored each sample on the same 5-point rubric used by the LLM judges. Fleiss' κ across the five annotators is 0.72 (NQP-Priv), 0.75 (NQP-Wild), and 0.70 (NQP-Share), indicating substantial agreement and confirming that the rubric yields consistent human judgments.

Human–LLM correlation. We compute Spearman's ρ between the per-sample LLM-judge score (majority vote of three judges) and the corresponding human score (majority vote of five annotators) on the shared 250-sample subsets. The correlation is 0.78 (NQP-Priv), 0.81 (NQP-Wild), and 0.76 (NQP-Share), showing strong sample-level agreement beyond the ranking-level consistency reported in §4.2.

LLM inter-judge agreement. Fleiss' κ among the three LLM judges (Claude Opus 4.7, GPT-5.5, Gemini-3.1-Pro) is 0.68 (NQP-Priv), 0.71 (NQP-Wild), and 0.66 (NQP-Share). The moderate-to-substantial agreement supports the use of majority vote to aggregate their ratings.

Human validation of benchmark curation. Since the curation pipeline relies on LLM judgments to assess predictability, we validate that retained samples reflect genuine predictability rather than LLM-specific biases. Three human annotators independently reviewed 400 randomly sampled instances from the curation candidate pool (200 retained KEEP instances and 200 rejected DROP instances) and judged whether the target query was predictable from the conversation context. Human–LLM agreement on the KEEP/DROP decision reached 92% (Cohen's $\kappa = 0.83$), confirming that the LLM cascade reliably identifies predictable conversations. Furthermore, the evaluation judges (Claude, GPT-5.5, Gemini-3.1-Pro) are largely distinct from the primary curation model (GLM-5), and Gemini serves only as a secondary reviewer in Stage III, limiting the overlap between data selection and scoring.

J Score Distribution Analysis

Table 12 shows the full rubric-level score distribution on NQP-Wild using the recursive memory interface, comparing an untrained base model against the RL-trained model and a frontier API.

Model	1	2	3	4	5	Mean
Base Qwen	26	24	25	16	9	39.57
RL Qwen	21	22	23	20	14	46.00
Gemini-3.1-Pro	15	22	25	22	16	50.72

Table 12: Score distribution (%, rounded) by rubric level on NQP-Wild. All rows use recursive memory. 1 = Irrelevant, 2 = Slightly related, 3 = Topic related, 4 = Highly aligned, 5 = Intent hit.

RL training shifts the distribution markedly toward higher rubric levels. Compared to the untrained base, score ≥ 4 predictions (Highly aligned or Intent hit) rise from 25% to 34% ($+9$ pp), while score ≤ 2 predictions drop from 50% to 43% (-7 pp). Gemini-3.1-Pro extends this trend further, reaching 38% at score ≥ 4 and only 37% at score ≤ 2. The moderate overall mean (~ 50) does not reflect uniformly mediocre predictions; rather, a substantial fraction of predictions are actionable (score ≥ 4), and the mean is pulled down by genuinely difficult samples that remain in the lower tiers. We note that open-ended next-query prediction is inherently harder than constrained tasks such as next-utterance selection or slot filling, because any turn in a multi-turn conversation may branch into multiple plausible directions. The 41%–52% of Hard samples in NQP-Bench (Figure 8) further depress absolute scores. Within this challenging setting, the consistent relative gains of RL training ($+6.4$ over Base) and our method over Full-history demonstrate meaningful progress.

From a deployment perspective, the 34–38% of predictions reaching score ≥ 4 are directly useful for downstream applications such as follow-up suggestion, pre-fetching retrieval results, and speculative response generation. Even score-3 predictions, which capture the correct topic but miss the exact question, can inform coarse-grained precomputation (e.g., warming a relevant document cache). Because the predictor runs asynchronously while the user is still reading the current response, even partially correct predictions incur no user-facing latency cost and can be silently discarded when inaccurate.

K Fine-Grained Evaluation and Error Analysis

To gain a comprehensive understanding of our method, we first stratify performance by difficulty and intent transfer paradigm (Table 13), then analyze the dominant error patterns.

Fine-grained performance breakdown. Aggregating across difficulty levels, our method outperforms Full-history on three of the four paradigms: Deepening ($+3.9$), Application ($+2.1$), and Challenge ($+3.0$), demonstrating broad effectiveness across diverse intent dynamics. The largest cell-level gains appear on Easy $\times$ Deepening ($+5.1$) and Hard $\times$ Challenge ($+5.7$), where the intent chain naturally tracks progressive exploration or unresolved tensions in the assistant's output. The sole exception is Associated Shift, where Full-history leads by $+3.3$ on Easy samples. This is expected: when the next query connects to non-adjacent earlier details through a grounded lateral association, the memory may have already compressed away the relevant earlier context, while raw history preserves these latent cross-topic links.

Difficulty	Method	Intent Transfer Paradigm			
		Deep.	Appl.	Assoc.	Chall.
Easy	Full-hist	45.1	42.0	**50.1**	44.3
	Ours	**50.2**	**44.6**	46.8	**45.1**
Hard	Full-hist	44.6	38.8	**44.5**	39.5
	Ours	**45.8**	**40.1**	42.7	**45.2**

Table 13: Performance (LLM Judge $\times 100$) stratified by difficulty and intent transfer paradigm on NQP-Wild (RL-trained Qwen). Bold indicates the better method.

Detail loss under compression. The bounded memory is optimized to retain high-level intent trajectories, but fine-grained details (e.g., specific numbers, exact constraints, or minor sub-topics) may be compressed away if the model judges them less salient at the time of the update. When the user's next query happens to depend on such details, prediction quality suffers. This pattern is most pronounced on Easy samples with Associated Shift, where Full-history outperforms our method by $+3.3$ (50.1 vs. 46.8 in Table 13), because the relevant detail often originates from a non-adjacent earlier turn that the memory has already summarized away.

Error distribution. Across all Hard samples on NQP-Wild, our method produces scores ≤ 2 (Slightly related or Irrelevant) on approximately 18% of instances. Qualitative inspection reveals two dominant patterns: (1) *detail-loss errors*, where the memory captures the overall intent trajectory but drops a specific detail that the next query depends on; and (2) *ambiguity errors*, where the conversation state genuinely supports multiple plausible next queries and the model's top candidates diverge from the ground truth despite being contextually reasonable.