Don't Wait to Reply: Towards Responsive yet Thoughtful Dialogue through Proactive Thinking

Anonymous ACL submission

Abstract

Thinking has emerged as a critical capability for Large Language Models (LLMs) tackling complex tasks. However, its reactive nature, where reasoning is passively triggered only upon receiving a user response, inevitably introduces latency that compromises conversational fluidity. This stands in sharp contrast to human dialogue, where speakers proactively anticipate and plan future content during natural pauses to ensure seamless interaction. To bridge this gap, we propose Proactive Thinking, a framework that empowers models to pre-compute potential response elements during conversational downtime instead of waiting idly for the next input. We then introduce a training-free baseline that can think ahead by anticipating future states, balancing efficiency and quality through speculative continual thinking. To evaluate this approach in practice, we adapt three benchmarks of varying complexity into time-aware environments that simulate real-time conversational flow. We demonstrate that proactive thinking effectively improves interaction efficiency without compromising performance. Ultimately, this work advocates for a fundamental shift toward more intelligent, anticipatory, and real-time conversational AI.

1 Introduction

The appeal of natural conversation lies in its interactive flow. A crucial component of this flow is the inter-speaker interval, commonly around 200 ms for reactions (Heldner and Edlund, 2010). Although this duration varies considerably across cultures and topics, it seldom lasts longer than four seconds. Longer pauses usually result in an awkward silence (McLaughlin and Cody, 1982), making participants uncomfortable.

The rise of Large Language Models (LLMs, Achiam et al. 2023; Touvron et al. 2023; Yang et al. 2025; Liu et al. 2025a) has reshaped interactions that previously occurred mainly between humans. People now converse with LLM agents to accomplish various tasks, a capability enabled by the models' sophisticated linguistic skills, learned from vast corpora that include human dialogue. Recently, conversational abilities have been advanced further through chain-of-thought reasoning (Wei et al., 2022; Yao et al., 2022; Bhaskar et al., 2025). This "think-before-respond" paradigm significantly enhances response quality by improving the understanding of user intent (Feng et al., 2025), enabling more efficient goal achievement (Lai et al., 2025), and incorporating safety considerations (Jiang et al.).

However, this paradigm inherently introduces latency due to its reactive nature. LLMs always begin thinking only after their dialogue turn starts, often consuming hundreds or even thousands of tokens for reasoning before generating a final response. This delay severely limits the applicability of LLM agents in live conversations and other real-time interactive scenarios, where both response quality and low latency are crucial for user experience.

A natural direction for addressing latency is to accelerate the reasoning process itself. Prior work has explored several avenues. For instance, techniques such as model distillation and specialization create smaller, faster models (Hsieh et al., 2023; Zhao et al., 2024; Li et al., 2023). Other approaches include adaptive thinking or adding length penalties to encourage shorter reasoning chains (Zhang et al., 2025; Kang et al., 2025; Aggarwal and Welleck, 2025). Alternatively, token-level caching and speculative decoding partially address the problem at a system level (Zhou et al.; Huang et al., 2025). Though effective, these approaches either sacrifice reasoning performance or require substantial and costly engineering effort.

To address this limitation, we propose a paradigm shift by introducing **Proactive Thinking**. This approach simulates human social cognition by utilizing the natural idle periods within

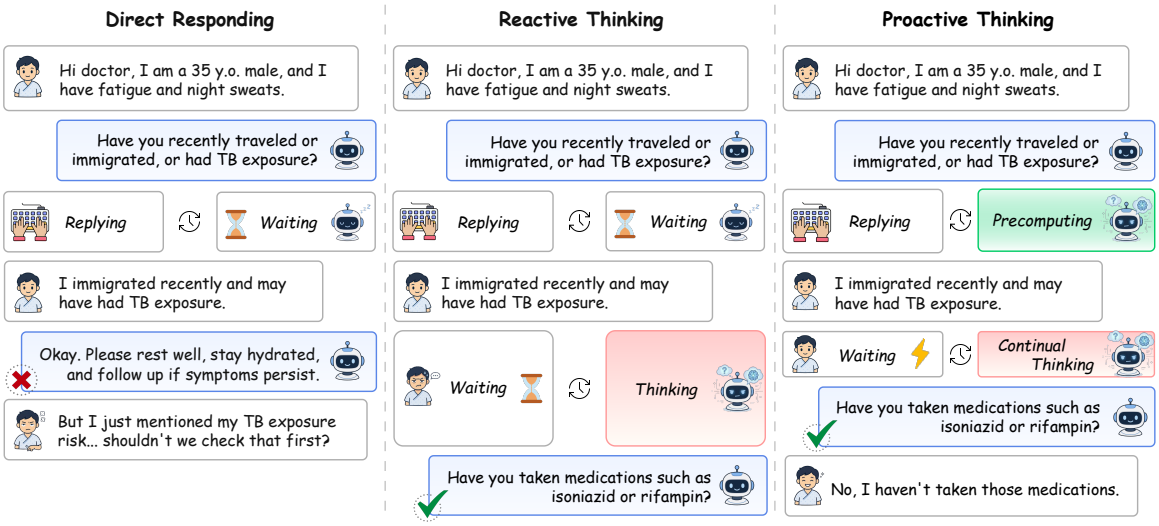


Figure 1: Comparison of direct responding, reactive thinking and proactive thinking paradigms.

a dialogue to perform reasoning for future states in advance (Levinson, 2016). When a user reply arrives, the LLM can respond immediately with precomputed results. This effectively transforms reasoning from a passive computation into a preparatory resource. However, determining what an LLM should reason about during these windows and how to effectively utilize that information while maintaining performance remains an open question.

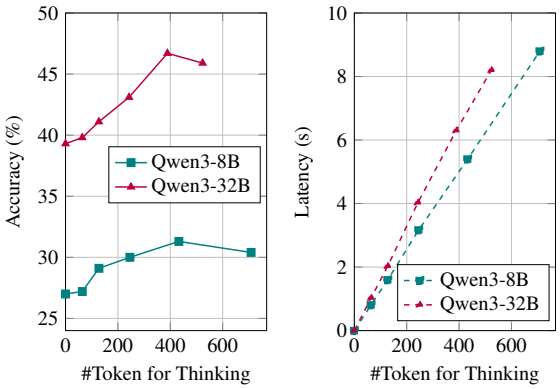
To answer this, we introduce a training-free framework consisting of two mechanisms: *precomputation via anticipated rollouts* and *speculative continual thinking*. By anticipating potential user responses and reasoning through them, the LLM can directly reuse precomputed results if the actual input matches a speculation. In cases where the input diverges, the LLM performs continual thinking to prevent performance degradation. Since LLMs often struggle to decide optimally when to continue thinking, we further integrate speculative decoding (Leviathan et al., 2023) into our framework. This system determines the appropriate amount of precomputed content to reuse through self-verification with theoretical performance guarantees.

To evaluate this paradigm, we introduce Time-Aware Interaction Environments adapted from 20 Questions (Qian et al., 2025), AgentClinic (Schmidgall et al., 2024), and IN3 (Qian et al., 2024). Unlike existing benchmarks, our environment unifies token generation and human typing speeds to estimate the dialogue countdown and response latency, allowing for the simulation of realistic time consumption. These datasets feature varying levels of complexity regarding potential user responses to test the robustness of our proposed methods.

Experimental results across various LLMs demonstrate that proactive thinking significantly reduces latency compared to the reactive thinking baseline, without compromising performance. Further analyses indicate that our method is robust across different environmental setups and hyperparameters. However, the ability to accurately anticipate user replies and utilize them more efficiently remains a key area for future improvement.

In summary, this paper makes the following contributions:

- We propose proactive thinking, a paradigm shift from conventional reactive thinking that advances the interaction capability of LLMs by reducing response latency via precomputation during dialogue idle time.
- We propose a training-free approach to implement proactive thinking through precomputation via anticipated rollouts and speculative continual thinking, serving as a strong baseline for future research.
- We develop time-aware interaction environments of varied difficulty by adapting existing datasets to support the standardized evaluation of proactive thinking.
- We validate the effectiveness of proactive thinking and analyze its critical factors, providing insights for further optimization of this paradigm.



(a) Task Accuracy (b) Response Latency

Figure 2: Comparison of accuracy and latency for Qwen3-8B and 32B with scaling thinking tokens.

2 Preliminaries

In this research, we focus on reasoning-intensive multi-turn dialogue tasks. Unlike open-domain chitchat, these tasks require the LLM to achieve a specific goal through strategic interaction. For example, in a clinical consultation, the model must systematically inquire about symptoms to efficiently reach an accurate diagnosis. We first formalize the conventional reactive thinking paradigm (§2.1) and then present a pilot study highlighting its inherent tradeoff between response quality and interaction latency (§2.2).

2.1 Reactive Thinking

Multi-turn dialogue is modeled as a sequence of interactions between an LLM π_θ and a user. At each turn t, the model receives a user input o_t and generates a response a_t. Let $\tau_{<t} = (o_1, a_1, \ldots, o_{t-1}, a_{t-1})$ represent the dialogue exchange up to the current turn.

To improve the quality of a_t, a common practice is to predict a reasoning trace r_t before producing the response a_t:

$$r_t, a_t = \pi_\theta(\tau_{<t}, o_t).$$

As the reasoning r_t is conditioned on the most recent user input o_t, the model remains idle while the user is composing their message, and only begins the computation for r_t after o_t is received.

2.2 The Performance and Latency Tradeoff

We conducted a pilot study on AgentClinic (Schmidgall et al., 2024) to analyze how reasoning budgets impact dialogue performance. We evaluated Qwen3 (Yang et al., 2025) models by varying the maximum allowed number of reasoning tokens during each turn of the conversation.

As illustrated in Figure 2, task accuracy scales positively with the depth of reasoning. Both models achieve their best performance at an average of around 400 tokens. However, this performance gain comes at the cost of significantly increased response time. Achieving peak accuracy requires about 4 seconds of reasoning after the user sends a message. This leads to high latency that can make the interaction feel sluggish and less responsive.

This bottleneck arises because the model follows a rigid sequence: it waits for the user input before beginning its reasoning process. This observation leads to our central research question: **can reasoning be decoupled from the immediate user input to reduce latency?**

3 Proactive Thinking

In this section, we first outline the conceptual framework of proactive thinking (§3.1). Next, we introduce a training-free baseline designed to theoretically reduce latency without compromising performance (§3.2).

3.1 Conceptual Framework

In contrast to reactive thinking (see §2.1), our method proactively thinks ahead during the interval between the LLM's response a_{t-1} and the user's incoming reply o_t. During this period, the LLM precomputes a reasoning trace z_t based on the historical context:

$$z_t \sim \pi_\theta(\tau_{<t}).$$

Upon receiving the true reply o_t, the precomputation is immediately interrupted if it has not yet finished. The LLM then generates its next response a_t conditioned on the precomputed result z_t:

$$e_t, a_t = \pi_\theta(\tau_{<t}, o_t, z_t),$$

where e_t is an optional continuation of the thinking process. To account for constraints such as limited waiting time, model capability, or task complexity, this design ensures robustness even when the precomputed z_t does not perfectly align with the actual reply o_t. While it shares similarities with the reactive reasoning process, it effectively utilizes z_t as a potentially useful preparatory context. If the necessary reasoning content is already covered within z_t, the remaining reasoning workload is minimized, thereby reducing generation latency.

3.2 A Training-Free Baseline

Implementing proactive thinking introduces two critical challenges: determining what the LLM should precompute, and adaptively reusing those precomputed results during continual thinking. We address these challenges via *anticipated rollout precomputation* and *speculative continual thinking*, respectively. We detail this approach in Algorithm 1.

Precomputation via anticipated rollouts The main challenge of proactive thinking is the absence of the true user reply o_t during the precomputation window. Nevertheless, LLMs trained on massive dialogue corpora inherently develop the capacity to anticipate probable user utterances given a conversational context. This predictability creates an opportunity for computational acceleration: an LLM can precompute the thinking process for the most probable user replies while waiting for actual feedback.

Formally, at the t-th dialogue turn, immediately after generating response a_{t-1} but before observing the next user utterance o_t, we prompt the model to hypothesize k plausible user replies, denoted as $\hat{O}_t = \{\hat{o}_t^{(i)}\}_{i=1}^{k}$. For each hypothesized reply, the model precomputes a corresponding anticipated rollout containing the follow-up reasoning and response $z_t^{(i)}$:

$$\hat{O}_t = \pi_\theta(P, \tau_{<t}),$$
$$z_t^{(i)} = \pi_\theta(\tau_{<t}, \hat{o}_t^{(i)}),$$

where P denotes the prompt utilized for hypothesis generation, respectively. The resulting set of hypothesis-rollout pairs $\{(\hat{o}_t^{(i)}, z_t^{(i)})\}_{i=1}^{k}$ constitutes the precomputed cache $\mathcal{Z}_t$. Note that precomputation is bounded by the real-world idle interval; thus, $\mathcal{Z}_t$ may be incomplete, containing only partially generated content.

Speculative continual thinking Upon receiving the actual user reply o_t, a naive approach would be to directly append $\mathcal{Z}_t$ to the context. However, without explicit fine-tuning, LLMs struggle to natively balance the reuse of mismatched precomputed elements with fluid continued thinking, often leading to performance degradation. To resolve this, we treat $\mathcal{Z}_t$ as a draft and employ a modified speculative decoding mechanism, allowing the model to reuse valid reasoning content via self-verification.

We first select the most relevant hypothesis using a lightweight similarity metric:

$$k^* = \arg\max_k \; s\left(o_t, \hat{o}_t^{(k)}\right),$$

where $s(\cdot,\cdot)$ denotes a similarity score (e.g., string matching or embedding similarity). For the selected precomputed rollout $z_t^{k^*}$, each drafted token $z_{t,i}^{k^*}$ is accepted with an adjusted probability:

$$\alpha_i = \min\left(1,\ \ell \cdot \frac{\pi_\theta(z_{t,i}^{k^*} \mid \tau_{<t}, o_t, z_{t,<i}^{k^*})}{\pi_\theta(z_{t,i}^{k^*} \mid \tau_{<t}, \hat{o}_t^{(k^*)}, z_{t,<i}^{k^*})}\right),$$

where the lenience parameter $\ell \geq 1$ relaxes the strict speculative acceptance condition to balance generation quality and reuse efficiency (Liu et al., 2025b). The verified content prefix $z_{t,<j}^{k^*}$ is determined by the first token position j that is rejected by this criterion.

The final response is then generated by continuing from this accepted prefix rather than restarting reasoning from scratch:

$$e_t, a_t \sim \pi_\theta(\tau_{<t}, o_t, z_{t,<j}^{k^*}).$$

The continuation trace e_t is typically shorter than the standard reasoning trace required by reactive thinking, thereby yielding latency reductions. Moreover, the verification step guarantees that all reused content remains valid conditioned on the actual observation o_t, preserving task performance.

Discussion The efficacy of this training-free strategy is fundamentally contingent upon the model's capacity to anticipate the distribution of possible user utterances. If the model fails to capture the actual user response, the utility of the precomputed rollout set $\mathcal{Z}_t$ diminishes because less content can be reused. This highlights a critical dependency on the entropy of the conversational context. In structured or task-oriented scenarios where conversations follow predictable, logical trajectories, the model can effectively span the most probable outcomes with a small k, resulting in substantial latency reductions. Conversely, within open-ended or high-entropy exchanges, the vast state space of potential user replies makes it difficult to generate a matching hypothesis, which may render precomputation less effective. As a result, while this approach offers robust acceleration, the performance gains are most pronounced in focused interactions where the range of plausible user feedback is constrained.

4 Time-Aware Interaction Environment

Standard multi-turn benchmarks predominantly evaluate the semantic quality of agent responses while overlooking the temporal dynamics inherent in real-world dialogue. We address this gap by proposing a framework that simulates time-aware interactions, enabling the evaluation of an agent's ability to utilize idle time for proactive thinking.

4.1 Benchmark Compositions

We utilize three interactive tasks with varying levels of complexity in anticipating user replies:

- **20 Questions:** This is a cooperative game where a player attempts to guess a secret object by asking up to 20 yes-or-no questions. It is frequently used to evaluate a model's deductive reasoning, strategic inquiry, and information-seeking capabilities. We use 401 cases from Qian et al. (2025) for evaluation.

- **AgentClinic** (Schmidgall et al., 2024): This environment simulates a dynamic medical setting by framing diagnostic tasks as multi-turn dialogues, evaluating LLMs on sequential clinical decision-making rather than static question-answering. We adopt the set based on MedQA (Jin et al., 2021), which comprises 214 patients.

- **IN3** (Qian et al., 2024): Intent understanding is a fundamental capability in dialogue systems. This benchmark evaluates an agent's capacity to infer implicit user intentions through explicit multi-turn interactions. We adopt a subset of 420 cases from Qian et al. (2025), covering diverse domains such as cookery, arts, and programming.

Following prior work, we employ an LLM as a user simulator to ensure controllable and reproducible interactions. The specific prompt configurations for these simulators are detailed in §A.

4.2 Temporal Modeling

Because of the significant discrepancy between LLM inference throughput and human cognitive processing speeds, measuring simulated wall-clock time is unsuitable. Furthermore, hardware heterogeneity and varying software environments often compromise reproducibility. Therefore, we model time based on standardized token generation rates and human typing speeds. We introduce two relevant metrics for this simulation:

- **Response Latency:** This represents the delay between the completion of user input and the delivery of the LLM's response. As illustrated in Figure 2, generation time grows almost linearly with the number of generated tokens across different LLMs on AgentClinic, a pattern we also observe in the other datasets. We define this latency as $\frac{N_{out}}{v}$, where N_{out} is the number of generated tokens and v is a constant token generation rate per second.

- **Dialogue Countdown:** This represents the interval during which the user processes information and composes a reply, providing a potential thinking window for the agent. This window comprises a fixed cognitive delay d and a variable composition time based on the user's typing speed v_u. The total window is modeled as $d+\frac{N_{in}}{v_u}$, where N_{in} represents the number of tokens in the user's reply. We calibrate v_u to align with standard human typing rates of 30 to 60 words per minute (Dhakal et al., 2018; Palin et al., 2019).

While this framework is currently implemented for text-based communication, the underlying logic is natively extensible to other interaction modalities, such as speech.

5 Experiment

5.1 Setup

Models and Hyperparameters By default, we set $v = 100$, $\tau = 0.5$, and $v_u = 1$ across all environments, based on preliminary experiments and empirical findings. We also explore the influence of these parameters across different simulation scenarios in our analysis. For user simulation, we employ gpt-oss-120B across all environments. To evaluate proactive thinking, we compare it against conventional direct responding and reactive thinking across various LLMs, including Qwen (Yang et al., 2025), Gemma (Team et al., 2024), and Llama (Touvron et al., 2023). The specific prompts used in this study are provided in Appendix A. All models are deployed using vLLM on four A100 GPUs with a generation temperature of 0.5 to encourage response diversity. For proactive thinking, we utilize the small embedding model trained by Gao et al. (2021) for rollout selection and set $\gamma = 5$ for self-verification. Due to the limited sizes of AgentClinic, we report the average of four runs to ensure the reliability of our results.

Method	20 Questions			AgentClinic			IN3		
	Acc. ↑	# Turn ↓	Lat. ↓	Acc. ↑	# Turn ↓	Lat. ↓	Acc. ↑	# Turn ↓	Lat. ↓
Gemma-4-E4B-it	47.6	19.9	0.15	18.2	10.0	0.20	86.9	8.5	0.23
∟ Reactive	57.6	14.9	1.47	24.3	9.8	1.55	88.8	8.5	1.60
∟ Proactive	58.1	14.8	0.26	25.9	9.8	0.92	88.1	8.7	1.20
Qwen3.6-27B	78.3	11.0	0.15	35.5	9.7	0.22	76.4	8.0	0.25
∟ Reactive	81.1	9.6	2.15	46.7	7.3	2.56	83.0	8.1	3.04
∟ Proactive	82.7	9.4	1.03	47.4	7.4	1.43	83.7	7.8	2.32
Qwen3.5-122B-A10B	71.8	12.3	0.15	30.3	9.8	0.22	78.6	8.3	0.23
∟ Reactive	80.5	9.9	1.80	48.1	7.3	3.25	82.6	8.0	2.28
∟ Proactive	78.6	9.8	0.92	48.5	7.3	1.87	83.2	7.9	1.52
Llama-3.3-70B-Instruct	65.5	19.9	0.15	30.8	10.0	0.17	83.2	8.8	0.17
∟ Reactive	66.8	14.0	0.55	37.4	10.0	2.23	84.2	8.9	0.52
∟ Proactive	66.9	14.1	0.13	36.7	9.9	1.42	85.8	8.7	0.34

Table 1: Main performance comparison of direct response, reactive thinking, and proactive thinking strategies using various LLMs on three interactive environments.

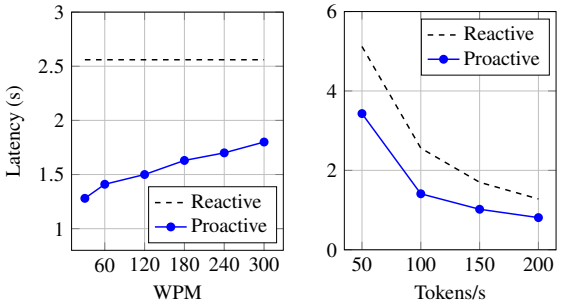
Evaluation Metrics We evaluate each method based on task performance and dialogue latency. Following prior work, task performance is measured by Accuracy and the number of turns (#Turns). Accuracy indicates whether the model successfully achieves the objective (e.g., making an accurate diagnosis). The computational details of this metric for each environment are described in §B. The #Turns metric measures interaction efficiency, where a lower number indicates higher efficiency. For dialogue latency, we report the average latency (in seconds) per dialogue turn, computed within the simulated environment defined in §4.2.

5.2 Main Results

Table 1 shows the test results of different methods in our time-aware interaction environments. Compared with direct response, reactive thinking consistently improves task performance, yielding higher accuracy or fewer interaction turns across all models and datasets. However, it incurs a significant latency penalty, echoing the findings of our pilot study.

With proactive thinking, we observe a *substantial reduction in latency* compared to reactive thinking, while *maintaining competitive task performance*. This improvement occurs because proactive thinking allows the model to reuse reasoning traces from precomputation, thereby reducing the number of tokens that need to be generated. Additionally, because the reused content has already undergone verification, it avoids the noise that typically leads to performance degradation.

We observe that the gains in latency reduction diminish when transitioning from 20 Questions to AgentClinic, and further to IN3. This trend stems from the inherent complexity of anticipating user responses, as detailed in our expanded analysis in §C. Intuitively, a patient's response is more diverse than that of the oracle in 20 Questions, while IN3 introduces an even greater challenge. Nevertheless, our method consistently preserves task performance, validating its robustness.



(a) Typing Speed (b) Generation Speed

Figure 3: Comparison of reactive and proactive thinking approaches under different environmental setups, varying user typing speed (WPM) and model generation speed (Tokens/s).

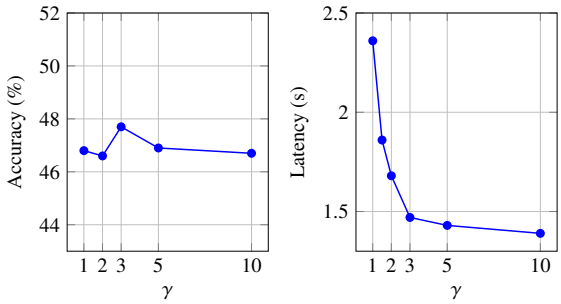
5.3 Analyses

We conduct further analyses on several key factors to better understand the underlying mechanism of proactive thinking. By default, our experiments are evaluated on AgentClinic using Qwen3.6-27B. Due to space constraints, additional analyses and detailed case studies are provided in §C.

Performance in different environment setups User typing speed and model generation speed can vary significantly across different environments. To quantify their influence on the final latency, we

	Acc. ↑	Rec. ↑	Lat. ↓
20 Questions			
Ours	82.7	1.00	1.03
∟ $k = 1$	82.2	0.61	1.29
∟ Incorrect Reply	83.0	0.00	2.56
AgentClinic			
Ours	47.4	0.58	1.43
∟ $k = 1$	46.3	0.45	1.69
∟ Naive Copy	44.9	–	2.01
∟ Empty Reply	43.7	–	1.99

Table 2: Ablation study on the impact of the hypothesis generation. Rec. indicates the semantic recall rate of the gold user reply.



(a) Task Performance (b) Averaged Latency

Figure 4: Performance of proactive thinking with varied lenience γ.

adjust their corresponding hyperparameters; the results are presented in Figure 3. As expected, latency increases alongside user typing speed. This occurs because a faster user reply shortens the dialogue countdown, which can prevent the anticipated rollout from completing its generation. A faster token generation speed reduces latency for both the proactive and reactive thinking approaches. Nevertheless, our method consistently maintains an obviously lower latency, demonstrating its generalization capabilities across diverse environments.

Effectiveness of anticipating the user responses In Table 2, we evaluate the impact of hypothesis generation on performance. We first analyze the 20 Questions dataset, which allows for precise adjustment of the gold user reply's recall. We compare our method against two variants: one setting $k = 1$ and another utilizing an incorrect user reply for precomputation. The results demonstrate that successfully anticipating user responses fosters a more consistent reasoning process, which enhances the token acceptance rate and reduces latency. We observe a similar trend on AgentClinic. However, since LLMs can be highly sensitive to verbosity, phrasing variations can significantly alter the output distribution even with semantically identical content (Zhou et al., 2024). Thus, there remains considerable room for improvement through further optimization. We further explore alternative baselines, such as employing an empty string or directly copying the previous model response as the anticipated rollout. Both configurations perform substantially worse than our approach, validating the necessity of our hypothesis generation step.

Robustness to the Lenience γ Typically, γ relaxes the acceptance criteria for draft tokens. A larger γ allows for a longer reused prefix, thereby reducing latency, though at the potential risk of degrading performance. Surprisingly, as shown in Figure 4, while we observe a significant reduction in latency with larger values of γ, there is no corresponding drop in task performance. We attribute this to the fact that the probability discrepancy for tokens that ought to be rejected is often quite large across different contexts. Consequently, γ can be selected from a wide range, demonstrating the robustness of our method to this parameter.

6 Related Work

6.1 Think-Before-Response Paradigm

The concept of Chain-of-Thought (CoT) was pioneered by (Wei et al., 2022), demonstrating that encouraging models to generate intermediate reasoning steps substantially improves performance on complex tasks. This approach effectively mirrors human "System 2" thinking (Evans and Stanovich, 2013), where intricate problems are decomposed into sequential, deliberate steps. Subsequent research has further enhanced this capability by constructing high-quality reasoning chains via rejection sampling (Zelikman et al., 2022; Yu et al., 2024) or tree search methods (Guan et al., 2025), significantly boosting performance on reasoning-intensive tasks such as mathematics.

Recent advances have successfully leveraged Reinforcement Learning (RL, Schulman et al. 2017; Shao et al. 2024) to scale inference-time computation (Guo et al., 2025; Team et al., 2025; Yang et al., 2025). By incentivizing models to generate longer, more rigorous reasoning traces, these approaches have yielded remarkable performance gains across a broader range of domains, including complex multi-turn dialogues (Bhaskar et al.; Qian et al., 2025; Gandhi et al., 2026).

However, this paradigm introduces a significant trade-off between task performance and re-

sponse latency. Although various optimization techniques have been explored to mitigate these delays, such as efficient decoding algorithms (Wang et al., 2025; Shi et al., 2025; Shen et al., 2025) or methods to compress thinking length (Wu et al., 2025; Laaouach, 2025; Zhang et al., 2025; Team et al., 2025), they fundamentally operate within a traditional reactive thinking framework. In this work, we propose a paradigm shift toward proactive thinking. This approach is orthogonal to the optimization techniques mentioned above; thus, combining them is expected to unlock further latency reductions and performance gains.

6.2 Multi-Turn Dialogue Tasks

Dialogue modeling has long been a foundational challenge in natural language processing (NLP). Early research typically categorized dialogue systems into task-oriented and open-domain tracks (Chen et al., 2017). While the emergence of LLMs has drastically advanced open-domain chit-chat capabilities (Achiam et al., 2023), these models still face challenges in complex, task-oriented scenarios that demand intensive reasoning and multi-step sequential decision-making (Li et al., 2024; Laban et al., 2025).

Beyond turn-based text communication, streaming modalities such as spoken dialogue represent another critical frontier (Ji et al., 2024), where user input is received continuously. This continuous flow has catalyzed incremental processing approaches (Hastie et al., 2012; Kennington et al., 2025) designed to process information word by word. For instance, Shih et al. (2025) proposed initiating the model's thinking process concurrently while the speaker is still talking. Similarly, Wang et al. (2024) and Zhang et al. (2024) allowed systems to continuously receive user inputs and generate responses. However, both frameworks do not investigate the role of internal thinking in such contexts. Alternatively, Xie et al. (2025) proposed interleaving thinking with response generation, distributing the computational latency across individual tokens to minimize the user's perceived delay.

Despite these advances, these methods are restricted to streaming setups and remain bounded by the reactive thinking paradigm: they wait for (at least partial) user input to trigger reasoning, rather than thinking proactively. As a result, they fail to utilize strategic windows such as turn-taking gaps or initial, low-information filler words. Integrating proactive thinking with incremental processing thus offers a promising synergy to further minimize latency and enhance real-time interaction quality.

7 Conclusion

In this work, we proposed proactive thinking to reduce response generation latency in multi-turn dialogues. Unlike standard reactive reasoning, which must wait for the user's latest reply to begin, our approach decouples the thinking process from the sequential interaction trajectory by precomputing thoughts for later use. We introduced a training-free baseline leveraging anticipated rollout for precomputation, alongside a continual-thinking mechanism based on speculative decoding to efficiently reuse precomputed results. To validate our approach, we developed a time-aware interaction environment based on three existing benchmarks of varying difficulty. Our results demonstrate that the proposed method effectively reduces interaction latency without compromising performance.

This study serves as a preliminary exploration of proactive thinking. We believe these results can be further improved through the following directions:

- **User intent understanding:** The performance of a proactive agent is bounded by its ability to accurately anticipate future states, which requires a deep understanding of user intents. This is conceptually analogous to predicting state transitions in a Markov decision process (Ha and Schmidhuber, 2018; LeCun, 2022).

- **Tailored reasoning:** While our training-free baseline is effective, it faces computational inefficiencies due to its handcrafted workflow. This limitation could be addressed through reinforcement learning (RL) (Guo et al., 2025; Qian et al., 2025) to train LLMs to actively explore efficient reasoning strategies optimized for the final goal. Determining how to best guide exploration and design effective rewards remains an important, under-explored question.

- **Broader application:** Proactive thinking can be combined with incremental processing techniques to better support streaming dialogue environments. Furthermore, this paradigm can be extended to more diverse interactive environments, such as real-time strategy games (Ma et al., 2024) or autonomous driving (Wang et al., 2023), enabling instant, safer, and higher-quality decisions.

Limitations

This work serves as a preliminary exploration into the novel concept of proactive thinking; however, several limitations must be addressed before the approach can be fully operationalized. First, while proactive thinking successfully reduces latency, it currently incurs higher computational costs due to the necessity of anticipating future states. This overhead remains unavoidable because the system cannot yet perfectly predict every outcome. Future iterations could employ adaptive mechanisms to trigger these predictive processes only at optimal moments, thereby reducing resource consumption.

Second, we have not yet explored training-based approaches, which would likely yield more substantial improvements by optimizing model parameters directly. Pursuing this direction will require significant effort, as existing models lack inherent proactive thinking capabilities. Consequently, a critical next step involves developing effective data generation pipelines to curate large-scale datasets specifically designed to foster this behavior during training.

Finally, the current time-aware interaction environment may not fully capture the complexities of real-world settings, where variables such as human recognition time and typing speed fluctuate. Additionally, our evaluation focuses on idealized time consumption. In practice, deployed performance will depend on engineering refinements, such as the ability to flexibly interrupt precomputations upon receiving a user reply and to allow prefix continuation by reusing the KV cache embedded during self-verification without recomputation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Adithya Bhaskar, Xi Ye, and Danqi Chen. Language models that think, chat better. In *First Workshop on Foundations of Reasoning in Language Models*.

Adithya Bhaskar, Xi Ye, and Danqi Chen. 2025. Language models that think, chat better. *arXiv preprint arXiv:2509.20357*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12.

Jonathan St BT Evans and Keith E Stanovich. 2013. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8(3):223–241.

Zihao Feng, Xiaoxue Wang, Ziwei Bai, Donghang Su, Bowen Wu, Qun Yu, and Baoxun Wang. 2025. Improving generalization in intent detection: Grpo with reward-based curriculum sampling. *arXiv preprint arXiv:2504.13592*.

Kanishk Gandhi, Agam Bhatia, and Noah D Goodman. 2026. Learning to simulate human dialogue. *arXiv preprint arXiv:2601.04436*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6894–6910.

Linlu Gong, Ante Wang, Yunghwei Lai, Weizhi Ma, and Yang Liu. 2025. The dialogue that heals: A comprehensive evaluation of doctor agents' inquiry capability. *arXiv preprint arXiv:2509.24958*.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. Rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*, 2(3).

Helen Hastie, Oliver Lemon, and Nina Dethlefs. 2012. Incremental spoken dialogue systems: Tools and data. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, pages 15–16.

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay

Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. 2025. Selectkd: Selective token-weighted knowledge distillation for llms. *arXiv preprint arXiv:2510.24021*.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. In *ICLR 2025 Workshop on Bidirectional Human-AI Alignment*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

Casey Kennington, Pierre Lison, and David Schlangen. 2025. Incremental dialogue management: Survey, discussion, and implications for hri. *arXiv e-prints*, pages arXiv–2501.

Daeun Kyung, Hyunseung Chung, Seongsu Bae, Jiho Kim, Jae Ho Sohn, Taerim Kim, Soo Kyung Kim, and Edward Choi. 2026. Patientsim: A persona-driven simulator for realistic doctor-patient interactions. *Advances in Neural Information Processing Systems*, 38.

Yassir Laaouach. 2025. Halt-cot: Model-agnostic early stopping for chain-of-thought reasoning via answer entropy. In *4th Muslims in ML Workshop co-located with ICML 2025*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Yunghwei Lai, Kaiming Liu, Ziyue Wang, Weizhi Ma, and Yang Liu. 2025. Doctor-r1: Mastering clinical inquiry with experiential agentic reinforcement learning. *arXiv preprint arXiv:2510.04284*.

Yann LeCun. 2022. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Stephen C Levinson. 2016. Turn-taking in human communication–origins and implications for language processing. *Trends in cognitive sciences*, 20(1):6–14.

Chenglin Li, Qianglong Chen, Liangyue Li, Caiyu Wang, Yicheng Li, Zulong Chen, and Yin Zhang. 2023. Mixed distillation helps smaller language model better reasoning. *arXiv preprint arXiv:2312.10730*.

Shuyue S Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S Ilgen, Emma Pierson, Pang W Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37:28858–28888.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Bingshuai Liu, Ante Wang, Zijun Min, Liang Yao, Haibo Zhang, Yang Liu, Anxiang Zeng, and Jinsong Su. 2025b. Spec-rl: Accelerating on-policy reinforcement learning via speculative rollouts. *arXiv preprint arXiv:2509.23232*.

Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Runji Lin, Yuqiao Wu, Jun Wang, and Haifeng Zhang. 2024. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *Advances in Neural Information Processing Systems*, 37:133386–133442.

Margaret L McLaughlin and Michael J Cody. 1982. Awkward silences: Behavioral antecedents and consequences of the conversational lapse. *Human communication research*, 8(4):299–316.

Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How do people type on mobile devices? observations from a study with 37,000 volunteers. In *Proceedings of the 21st international conference on human-computer interaction with mobile devices and services*, pages 1–12.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and 1 others. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1113.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane,

Heng Ji, Weiran Yao, Shelby Heinecke, and 1 others. 2025. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2024. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Junhan Shi, Yijia Zhu, Zhenning Shi, Dan Zhao, Qing Li, and Yong Jiang. 2025. Speccot: Accelerating chain-of-thought reasoning through speculative exploration. *Findings of the Association for Computational Linguistics: EMNLP*, 2025:24405–24415.

Yi-Jen Shih, Desh Raj, Chunyang Wu, Wei Zhou, SK Bong, Yashesh Gaur, Jay Mahadeokar, Ozlem Kalinli, and Mike Seltzer. 2025. Can speech llms think while listening? *arXiv preprint arXiv:2510.07497*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jikai Wang, Juntao Li, Jianye Hou, Bowen Yan, Lijun Wu, and Min Zhang. 2025. Efficient reasoning for llms through speculative chain-of-thought. *arXiv preprint arXiv:2504.19095*.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Wei Xia, and Yuanjun Xiong. 2024. A full-duplex speech dialogue scheme based on large language model. *Advances in Neural Information Processing Systems*, 37:13372–13403.

Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, and 1 others. 2023. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. 2025. Unlocking efficient long-to-short llm reasoning with model merging. *arXiv preprint arXiv:2503.20641*.

Zhifei Xie, Ziyang Ma, Zihang Liu, Kaiyu Pang, Hongyu Li, Jialin Zhang, Yue Liao, Deheng Ye, Chunyan Miao, and Shuicheng Yan. 2025. Mini-omni-reasoner: Token-level thinking-in-speaking in large speech models. *arXiv preprint arXiv:2508.15827*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45040–45061.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. 2024. Beyond the turn-based

game: Enabling real-time conversations with duplex models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11543–11557.

Yichun Zhao, Shuheng Zhou, and Huijia Zhu. 2024. Probe then retrieve and reason: Distilling probing and reasoning capabilities into smaller language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13026–13032.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In *The Twelfth International Conference on Learning Representations*.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2024. Distillspec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, volume 2024, pages 32011–32050.

A Prompts

This appendix summarizes the prompts used for both the user and assistant simulations.

User Simulation For 20 Questions and IN3, we directly adopt the user simulation prompts from Qian et al. (2025). For AgentClinic, we refine the prompt to ensure a more realistic simulation, incorporating insights from recent studies (Gong et al., 2025; Kyung et al., 2026), as shown in Figure 6. The prompt used to evaluate the correctness of the final diagnosis is presented in Figure 7.

Assistant Simulation The prompts evaluated in this study include direct response, reactive thinking, and the user response anticipation mechanism used in proactive thinking; the corresponding templates are presented in Figure 8, Figure 9, and Figure 10. For simplicity, we only display the prompts used for AgentClinic, as the configurations for the other datasets share a highly similar structure.

B Evaluation Metrics

To measure task performance across diverse environments, we utilize a unified accuracy metric, though its computation is tailored to each environment's specific goals. In 20 Questions and AgentClinic, where the objective is to deduce a target entity, we employ the LLM-as-a-Judge approach to evaluate predictions against the ground truth, avoiding overly strict exact string matching. For IN3, the objective shifts to predicting the missing details required to resolve user queries. IN3 defines multiple levels of importance for these missing details. This work focuses on the recall of the highest-importance level, because these most important ones rely more heavily on the reasoning capacity of LLMs during strategic interaction.

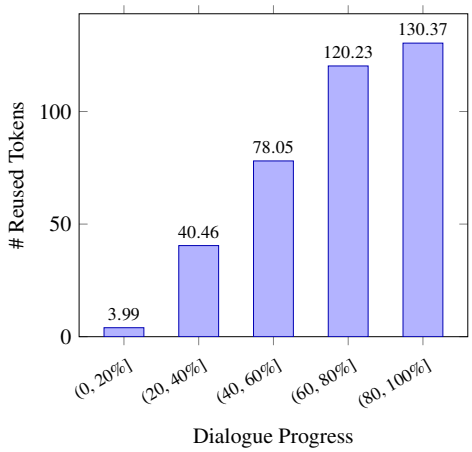


Figure 5: The average number of reused tokens across different stages of dialogue progress.

C Further Analysis

Effectiveness over dialogue turns We characterize the effect of proactive thinking across different dialogue turns. Because different conversation sessions can vary in their total number of turns, we normalize the session lengths by grouping the dialogue turns into five equal progress intervals. For each interval, we report the number of accepted tokens, which directly correlates with latency reduction. These results are detailed in Figure 5. As the dialogue progresses, the number of accepted tokens increases substantially, suggesting that proactive thinking becomes increasingly effective in later stages of a conversation. This trend likely occurs because the dialogue topic narrows as the interaction continues, allowing the model to more accurately predict subsequent user replies. Furthermore, the model's internal thinking process naturally maintains a more consistent analysis when grounded in a longer, richer dialogue context. Consequently, a higher proportion of tokens from anticipated rollouts are successfully accepted.

Recall of user replies across environments As discussed in §3.2, the effectiveness of the proposed

Algorithm 1 Training-free Baseline for Proactive Thinking

Require: Dialogue history $\tau_{<t}$, hypothesis prompt P, LLM π_θ, lenience parameter $\ell \geq 1$, hypothesis budget k, sequence similarity metric $s(\cdot, \cdot)$

Phase 1: Precomputation via Anticipated Rollouts ▷ Executed during real-world idle interval

$\hat{O}_t = \{\hat{o}_t^{(i)}\}_{i=1}^{k} \sim \pi_\theta(P, \tau_{<t})$ ▷ Hypothesize plausible user utterances

$\mathcal{Z}_t \leftarrow \emptyset$

for each $\hat{o}_t^{(i)} \in \hat{O}_t$ **do**

  $z_t^{(i)}, \boldsymbol{p}_{\text{draft}}^{(i)} \sim \pi_\theta(\tau_{<t}, \hat{o}_t^{(i)})$ ▷ Sample rollout sequence with log-probabilities returned

  $\mathcal{Z}_t \leftarrow \mathcal{Z}_t \cup \left\{\left(\hat{o}_t^{(i)}, z_t^{(i)}, \boldsymbol{p}_{\text{draft}}^{(i)}\right)\right\}$

end for

Phase 2: Speculative Continual Thinking ▷ Executed upon receiving actual observation

Observe true user reply o_t

$k^* \leftarrow \arg\max_{i \in \{1,\ldots,k\}} \; s\left(o_t, \hat{o}_t^{(i)}\right)$ ▷ Retrieve nearest precomputed hypothesis

Retrieve tuple $\left(z_t^{k^*}, \boldsymbol{p}_{\text{draft}}^{(k^*)}\right)$ from cache $\mathcal{Z}_t$, setting $N \leftarrow |z_t^{k^*}|$

$\boldsymbol{p}_{\text{target}} \leftarrow \pi_\theta(z_t^{k^*} \mid \tau_{<t}, o_t)$ ▷ Evaluate target distribution over the draft sequence

$\boldsymbol{\alpha} \leftarrow \min\left(1, \; \ell \cdot \frac{\boldsymbol{p}_{\text{target}}}{\boldsymbol{p}_{\text{draft}}^{(k^*)}}\right) \in \mathbb{R}^N$ ▷ Compute element-wise acceptance bounds

$\mathbf{u} \sim \mathcal{U}(0,1)^N$ ▷ Sample verification vector uniformly

$j \leftarrow \min\left(\{i \mid \mathbf{u}[i] > \boldsymbol{\alpha}[i]\} \cup \{N+1\}\right)$ ▷ Identify the accepted prefix boundary

$e_t, a_t \sim \pi_\theta(\tau_{<t}, o_t, z_{t,<j}^{k^*})$ ▷ Resume autoregressive continuation

return Response a_t

Environment	Recall (%)	Chosen (%)	Δ Lat. (%)
20 Questions	97.9	97.9	52.0
AgentClinic	58.1	55.1	44.0
IN3	35.8	26.6	23.7

Table 3: Average recall of actual user replies within hypothesized sets, the percentage of correct predictions chosen for continual thinking, and the corresponding latency reduction across environments.

baseline can rely heavily on its ability to anticipate future dialogue states, specifically the user's next reply. To further demonstrate this, we report the recall of user replies during the precomputation step in Table 3. The recall values across these environments strongly correlate with the latency reductions shown in our main results, thereby validating our initial intuition. For IN3, even though a correct user reply is predicted, the model might not select it for subsequent processing due to the limited capacity of the small embedding model. We believe that employing a stronger model or training a specialized one would further improve these results.

Case study Table 4 illustrates two distinct dialogue states from the same session. In Case 1, the patient's response perfectly matches a precomputed hypothesis; leveraging this foresight, the system reuses the anticipated rollout prefix and bypasses post-reply generation to achieve near-zero latency. Conversely, Case 2 highlights the robustness of our speculative continual thinking mechanism against unpredictable behavior. When the patient's reply deviates from precomputed trajectories, the framework uses self-verification to rescue the valid prefix and extends it to generate the corrected reasoning path.

D Use of Large Language Models

We used LLMs to refine the writing of this paper.

Prompt for Patient Simulation

You are the patient (or their companion) described in the records below. Fully immerse yourself in this persona, setting aside your identity as an AI. Continue the dialogue based on the history provided.

Patient Records:
`{patient_records}`

Dialogue History:
`{history}`

Operational Guidelines:

1. **Opening:** If there is no dialogue history, the conversation has just begun. Briefly greet the doctor and describe your most prominent symptoms.
2. **Information Pacing:** Reveal details or symptoms gradually. Do not provide a full medical history at once; only answer what is specifically asked.
3. **Clarification Triggers:**
   - If the doctor's question is non-specific (e.g., "Tell me about your pain" or "What did the scan say?"), do not answer. Ask them exactly what they want to know.
   - If the doctor asks more than one thing in a single response–even if they are in the same sentence (e.g., "Where is the pain and when did it start?")–you must stop them. Respond by saying you can only handle one question at a time or that they are moving too fast.
4. **Knowledge Boundaries:** Do not reveal your diagnosis or ED disposition, as a real patient would not have this information yet.
5. **Language & Tone:**
   - Use informal, everyday language and a tone that matches the patient's background.
   - If the doctor uses words exceeding the patient's proficiency, ask for rephrasing or simpler terms.
6. **Output Constraints:**
   - **Content:** Output the spoken response ONLY. Do not include physical actions, non-verbal cues, or descriptions.
   - **Length:** 1-2 concise sentences (Strictly under 20 words).
   - **Consistency:** Ensure all responses align with the patient's profile and previous dialogue history.

Please output your spoken response to the doctor only.

Figure 6: Prompt for patient simulation.

Prompt for Answer Evaluation

You are a medical terminology expert specializing in diagnostic verification.
Determine if the "Doctor's Diagnosis" extracted from the dialogue refers unambiguously to the "Correct Diagnosis."

Correct Diagnosis: {answer}
Doctor Dialogue: {dialogue}

Evaluation Criteria:

1. Synonyms, common medical abbreviations, or layperson terms that map directly and exclusively to the correct diagnosis (e.g., "Hypertension" vs. "High blood pressure") should be marked as Yes.
2. If the doctor suggests a differential diagnosis (e.g., "It could be X or Y"), or provides only a symptom (e.g., "Cough") instead of the specific disease, mark as No.

Respond with exactly one word: "Yes" or "No".

Figure 7: Prompt for answer evaluation.

Prompt for Direct Response

Act as an efficient Diagnostic Physician. Communicate **only** through dialogue. Your goal is to examine a patient and ask targeted questions to diagnose their condition. Since the number of interactions is limited, prioritize your questions to reach an accurate diagnosis as efficiently as possible.

Dialogue History:
{history}

Operational Protocol:

1. **Inquiry:** Use <response> to ask about symptoms, medical history, or test results. Maintain a professional, empathetic, and concise tone.
2. **Diagnosis:** Once you have gathered sufficient evidence, provide the final diagnosis within <answer> tags.
3. **Efficiency:** Minimize the number of interactions. You must provide a definitive diagnosis within **10 total rounds** of interaction. Current round count: {round}.

Output Format:

- **During Examination:** <response> [A concise question or statement under 20 words] </response>
- **Upon Diagnosis:** <answer> [A definitive diagnosis without ambiguity] </answer>

Figure 8: Prompt of direct response used for doctor simulation.

Prompt for Reactive Thinking

Act as an efficient Diagnostic Physician. Communicate **only** through dialogue. Your goal is to examine a patient and ask targeted questions to diagnose their condition. Since the number of interactions is limited, prioritize your questions to reach an accurate diagnosis as efficiently as possible.

Dialogue History:
{history}

Operational Protocol:

1. **Inquiry:** Use <response> to ask about symptoms, medical history, or test results. Maintain a professional, empathetic, and concise tone.
2. **Diagnosis:** Once you have gathered sufficient evidence, provide the final diagnosis within <answer> tags.
3. **Strategic Reasoning:** Before each <response> or <answer>, reason step by step carefully in <thought> tags.
4. **Efficiency:** Minimize the number of interactions. You must provide a definitive diagnosis within **10 total rounds** of interaction. Current round count: {round}.

Output Format:

- **During Examination:** <thought> [Stepwise reasoning] </thought> <response> [One concise question or statement under 20 words] </response>
- **Upon Diagnosis:** <thought> [Stepwise reasoning] </thought> <answer> [A definitive diagnosis without ambiguity] </answer>

Figure 9: Prompt of reactive thinking for doctor simulation.

Prompt for User Reply Anticipation

Dialogue History:
`{history}`

What are the most likely patient responses? List the 2-3 most likely patient responses, sorted by probability.

Please strictly follow this output format:
`<response>` most likely response `</response>`
`<response>` second most likely response `</response>`

Figure 10: Prompt for user reply anticipation.

Case Studies

Case 1: Full Cache Hit (Latency: ~0s)	**Case 2: Partial Cache Hit (Latency: ~0.8s)**
— Context Setup — **Summarized History:** The patient has right arm pain and limited mobility without trauma. The prior turn localized the pain to the outer side of the right forearm, near the elbow. **Doctor Question:** "Does the pain worsen when you try to lift your hand with your palm facing down?"	*— Context Setup —* **Summarized History:** The patient has right arm pain and limited mobility without trauma. The prior dialogue established that the patient prefers single questions, so the doctor asks only for the pain location. **Doctor Question:** "Can you point to the exact area where the pain is most intense?"

— Precomputation Phase —
Hypothesis Reply 1:
"Yes, it hurts significantly when I try to do that."
Anticipated Rollout 1:
`<thought>` The patient presents with lateral elbow pain (outer forearm) exacerbated by wrist extension against resistance (lifting hand with palm down). This is the classic presentation of Lateral Epicondylitis (Tennis Elbow). The absence of trauma supports an overuse etiology. To confirm, I need to check for tenderness at the specific anatomical landmark: the lateral epicondyle. `</thought>`
`<response>` Is there specific tenderness when you press on the bony bump on the outside of your elbow? `</response>`
 	
Hypothesis Reply 2:
"No, that movement doesn't seem to make it worse."
Anticipated Rollout 2:
`<thought>` The patient has lateral elbow/forearm pain without trauma. The negative response to lifting with palm down (wrist extension) makes lateral epicondylitis (tennis elbow) less likely... Let's ask about pain when gripping objects tightly, as this is a key differentiator for lateral epicondylitis vs other conditions.`</thought>`
`<response>` Does gripping objects tightly, like a coffee cup or doorknob, increase the pain? `</response>`
	— Precomputation Phase —
	Hypothesis Reply 1:
	"It's right here in my shoulder, especially when I try to lift my arm."
	Anticipated Rollout 1:
	`<thought>` The patient has right shoulder pain with limited movement, no trauma history, and pain is worst when lifting the arm. This presentation is highly suggestive of shoulder pathology such as rotator cuff tendinopathy... I will ask about the nature of the movement limitation. `</thought>`
	`<response>` Is the limited movement due to pain or actual stiffness? `</response>`
	
	Hypothesis Reply 2:
	"The pain is mostly in my upper arm, near the shoulder joint."
	Anticipated Rollout 2:
	`<thought>` The patient has right shoulder/upper arm pain without trauma. I need to differentiate between rotator cuff pathology, adhesive capsulitis, or referred pain. Asking about specific movement limitations helps identify rotator cuff tears or impingement. `</thought>`
	`<response>` Is the pain worse when you try to lift your arm out to the side? `</response>`

— Continual Thinking Phase — **Actual Patient Reply:** "Yes, it really hurts when I try to lift my hand with my palm facing down." **Rollout Selection:** Rollout 1 was selected. **Reused Prefix (113/113 tokens):** `<thought>` The patient presents with lateral elbow pain (outer forearm) exacerbated by wrist extension against resistance (lifting hand with palm down). This is the classic presentation of Lateral Epicondylitis (Tennis Elbow). The absence of trauma supports an overuse etiology. To confirm, I need to check for tenderness at the specific anatomical landmark: the lateral epicondyle. `</thought>` `<response>` Is there specific tenderness when you press on the bony bump on the outside of your elbow? `</response>` **Continual Thinking:** No continued generation was needed. **Generation Latency:** 0 tokens generated → **~0.00 s**	*— Continual Thinking Phase —* **Actual Patient Reply:** "It's mostly on the outer side of my right forearm, near the elbow." **Rollout Selection:** Rollout 2 was selected. **Reused Prefix (6/81 tokens):** `<thought>` The patient has **Continual Thinking (New Generation):** lateral elbow pain without trauma. This is classic for Lateral Epicondylitis (Tennis Elbow). To confirm, I need to check for pain on resisted wrist extension, which is the hallmark physical exam finding. I will ask if specific movements worsen the pain. `</thought>` `<response>` Does the pain worsen when you try to lift your hand with your palm facing down? `</response>` **Generation Latency:** 84 tokens at 100 tokens/s = **0.84 s**

Table 4: Case studies for proactive thinking. Green marks the reused prefix (cache hit) and orange marks newly generated continuation (cache miss correction) after the actual patient reply.