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

Anonymous ACL submission

Abstract

While chain-of-thought reasoning has significantly advanced the reasoning capabilities of Large Language Models (LLMs), its sequential reactive thinking nature introduces substantial latency, which can degrade the responsiveness and fluidity of conversational systems. In this position paper, we propose proactive thinking—a paradigm inspired by human conversational dynamics in which LLMs perform reasoning during natural dialogue intervals or while the user is speaking. This approach enables the model to pre-plan elements of its response before its turn begins. We argue for its feasibility from a predictive branching perspective. As preliminary validation, we implement a prototype proactive-thinking system using prompting, demonstrating that LLMs can effectively plan upcoming responses in task-oriented dialogues, even without knowing the user's next utterance. Our work advocates for a shift toward more intelligent, real-time interaction models in future 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. 2025) 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. 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., 2025a; 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 fundamental limitation, this position paper proposes a paradigm shift by introducing **Proactive Thinking**. While conventional reactive thinking starts only after the latest user input, our approach decouples deep reasoning from the sequential interaction trajectories. It utilizes the natural idle periods within a dialogue to perform reasoning for future states in advance. Consequently, when a user reply arrives, the agent can

respond immediately with precomputed results, effectively cutting latency. In effect, this transforms reasoning from a passive, reactive computation into a proactive, preparatory resource.

The remainder of this paper is organized as follows. First, we introduce the conventional reactive thinking paradigm and conduct a pilot study. Second, we formalize the concept and mechanism of proactive thinking, arguing its feasibility through *predictive branching* perspective. Third, we implement a simple baseline system for proactive thinking and evaluate it on typical goal-oriented dialogue tasks, including TelepathyGym (Qian et al., 2025) and AgentClinic (Schmidgall et al., 2024), across Qwen3 (Yang et al., 2025) and DeepSeek-V3.2 (Liu et al., 2025) to validate its effectiveness. Finally, we discuss limitations and outline future research directions.

2 Preliminaries

In this research, we focus on goal-oriented dialogue tasks, which are inherently interactive decision-making problems. Unlike open-domain chit-chat without a specific objective, these tasks require an agent to reason about what to say in order to accomplish a final goal efficiently. For example, in a clinical inquiry, a physician must strategically ask questions to understand a patient's condition for making an accurate diagnosis. This makes reasoning an effective technique for improving performance. In this section, we first introduce the conventional *reactive thinking* paradigm (§2.1) and then present a pilot study examining its impact on both task performance and dialogue latency (§2.2).

2.1 The Reactive Thinking Paradigm

We model the interactive decision-making process as a partially observable sequential decision problem. At each step t, an LLM-based agent π_θ interacts with an environment by producing an action a_t and subsequently receives an observation o_t from the environment.

Let x denote the initial task description, and let $\tau_{<t} = (a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ represent the interaction history up to step $t-1$. The agent's objective is to choose actions that lead to a final decision y, such as an answer, a diagnosis, or task completion, that maximizes a task-specific utility.

To select an optimal action at each step, a common practice is to first predict a reasoning trace r_t before producing the action (Yao et al., 2022). This reasoning is internal and not directly observable to the environment. Formally, the agent follows the synchronous pattern:

$$r_t, a_t = \pi_\theta(x, \tau_{<t}), \tag{1}$$

where the reasoning r_t typically conditions on the most recent observation o_{t-1}, and the action a_t is generated afterward.

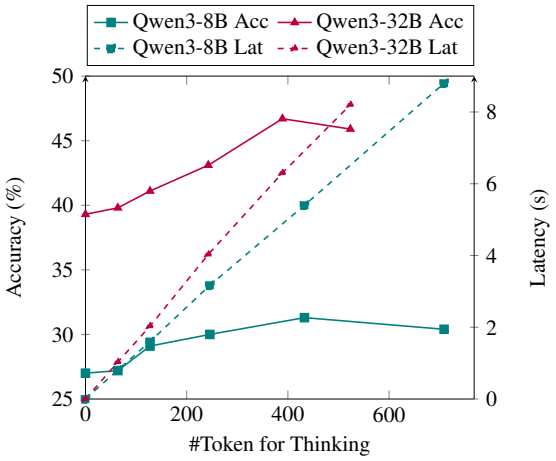

Figure 1: A comparison of the accuracy (left y-axis, solid line) and latency (right y-axis, dashed line) for Qwen3-8B and Qwen3-32B across maximum token limits of 0, 64, 128, 256, 512, and unlimited.

2.2 The Performance-Latency Trade-off

We conduct a pilot study to examine how the amount of reasoning affects performance and interaction efficiency under the conventional reactive thinking paradigm. Our experiments use AgentClinic (Schmidgall et al., 2024), a clinical inquiry dataset where an LLM agent dialogues with a simulated patient to reach a diagnosis. A detailed description of the setup is provided in §4.2.

We evaluate Qwen3-8B and Qwen3-32B (Yang et al., 2025) in their thinking mode, which allows the maximum number of reasoning tokens to be controlled. To enforce the reasoning budget, we insert the special token `</think>` to terminate the reasoning process once the token limit is reached, forcing the model to produce its final answer.

Figure 1 shows the average diagnosis accuracy and interaction latency per dialogue turn. We observe that task performance improves consistently as the reasoning budget increases. The number of dialogue rounds are uniformly around 5, indicating improved interaction efficacy. When reasoning is unrestricted, we observe a slight perfor-

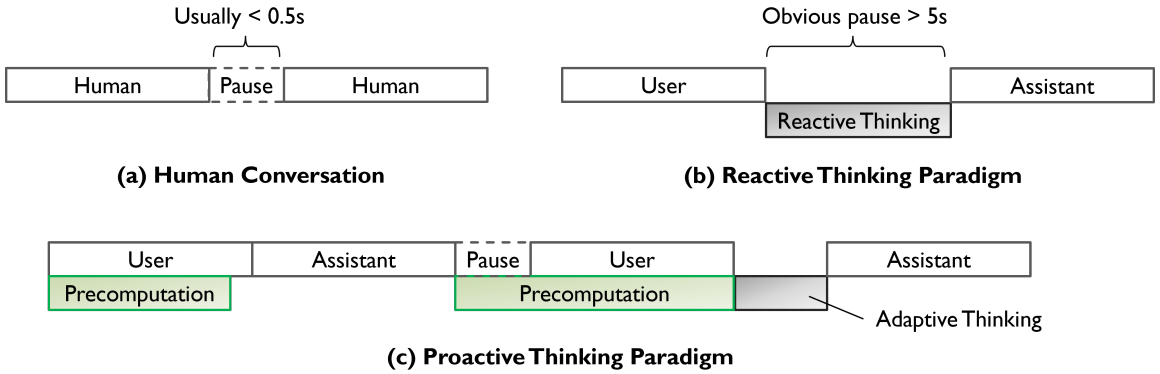

Figure 2: A comparison of reactive and proactive thinking paradigms. Proactive thinking uses precomputation to prepare results ahead of time, minimizing latency. If needed, on-demand continued thinking can be adaptively invoked to guarantee response quality.

mance decrease for both models, which can be due to the over-thinking (Chen et al., 2024). Nevertheless, Qwen3-32B still achieves a 6.6-point improvement in accuracy compared to its non-thinking mode, confirming the importance of reasoning for this task. However, interaction latency rises sharply with larger reasoning budgets. Both models achieved their best performance with latencies exceeding five seconds, raising concerns about awkward silences during interaction.

In summary, **the conventional reactive thinking paradigm improves task performance but incurs significantly higher latency**. This motivates the need for a more efficient approach. One contributing factor is its reactive nature: the agent must *wait* for the latest state (i.e., the user's response) before it can begin reasoning. This leads to our central research question: **can reasoning be decoupled from immediate feedback without sacrificing task performance?**

3 Proactive Thinking

In this section, we introduce proactive thinking, where an agent proactively thinks ahead for future dialogue turns without access to the latest observation (§3.1). We then argue, from a predictive branching perspective, that thinking can be decoupled from user feedback without a loss in performance (§3.2).

3.1 Precomputation for Latency Reduction

In contrast to the standard synchronous paradigm (see §2.1), the proactive thinking paradigm performs reasoning computations during idle periods. A common example is the interval between an agent's action a_t and the environment's subsequent observation o_t. Such idle windows occur naturally in applications like dialogue, where a user's response time allows for precomputation.

During this idle period, the agent precomputes a reasoning trace $\mathcal{R}_t$ based on the current context:

$$\mathcal{R}_t \sim \pi_\theta(x, \tau_{<t}, a_t).$$

Upon receiving the next true observation o_t, the agent updates its trajectory to $\tau_{<t+1}$. If the precomputation has not finished, it is forcibly stopped. The agent then generates its next action a_{t+1} conditioned on both the updated trajectory and the precomputed result:

$$a_{t+1} = \pi_\theta(x, \tau_{<t+1}, \mathcal{R}_t).$$

Due to constraints such as limited waiting time, model capability, or task complexity, the precomputed $\mathcal{R}_t$ may not contain reasoning suitable for the actual observation o_{t+1}. To ensure robustness especially in high-stakes tasks, the agent can perform adaptive continued thinking, which is similar to the reactive reasoning process but utilizes $\mathcal{R}_t$ as potentially useful preparatory context. In the worst case, where the precomputation offers no advantage, the proactive thinking approach incurs time consumption similar to that of the reactive paradigm.

Intuitively, the central challenge for proactive thinking lies in precomputation. This necessitates clarifying how precomputation is feasible and what form $\mathcal{R}_t$ could take to be useful for subsequent decision-making.

3.2 Feasibility: The Predictive Branching Perspective

In human conversation, we can often anticipate how the other person will respond. Similarly,

LLMs trained on massive dialogue corpora develop the capability to predict or reason about probable user utterances given a conversational context. We formalize this intuition through a predictive branching perspective.

We conceptualize each distinct possible user feedback as a *branch*. While the space of all linguistic branches is infinite, the probability mass in specific conversational states can be highly concentrated on just a few of them. For example, after a clinician asks, "*Is the pain sharp or dull?*", the responses "*sharp*", "*dull*", and "*neither*" capture the vast majority of the probability distribution over next utterances.

This predictability creates an opportunity for computational speedup. An LLM agent can precompute follow-up responses for the k most probable branches during the waiting time for the actual user feedback. If the received response corresponds to a prepared branch, the agent can reply instantly.

Formalization Let the LLM's prior predicted distribution over possible environment feedbacks (branches) be denoted P. The true distribution of actual feedback from the environment is Q. We assume the agent can prepare actions for the top-k branches ranked by P. The probability that the actual environment feedback $o_i \sim Q$ is among these pre-prepared branches is given by:

$$P_{\text{success}}(k) = \sum_{o_i \in \text{Top-}k(P)} Q(o_i),$$

where Top-$k(P)$ is the set of branches with the highest k probabilities under P.

Let T_a denote the time required for reactive reasoning per step. If adaptive continued thinking is performed, the expected latency per step under proactive thinking is then $T_a \cdot (1 - P_{\text{success}}(k)) \leq T_a$ and the task performance can be maintained.

Key Factors for Feasibility The efficacy of this precomputation strategy depends on two primary factors related to P:

- **Similarity of P and Q**: The strategy is viable only if the LLM's predicted distribution P closely aligns with the true environment feedback distribution Q. High similarity ensures that preparing for the branches deemed most likely by the LLM also covers the branches most likely to occur in reality.
- **Branching Entropy $H(P)$**: The entropy $H(P)$ measures the uncertainty or spread of the prediction. A low entropy state, where probability mass is concentrated on very few branches, is highly amenable to this approach. Conversely, in high-entropy, open-domain scenarios (e.g., after a question like "*What symptoms are you experiencing?*"), the user's next utterance is largely unconstrained. Here, even for moderate k, the chance $P_{\text{success}}(k)$ remains low, making precomputation inefficient.

Thus, the predictive branching perspective highlights that response precomputation is a feasible and beneficial optimization primarily in low-entropy, predictable dialogue states.

4 Experiment

While the concept of proactive thinking is intuitive, its implementation and verification require substantial effort. This involves jointly optimizing precomputation and adaptive thinking capabilities, as well as preparing an environment with human participants. In this study, to validate the feasibility of the proactive thinking paradigm, we first implement a simple version via prompting (§4.1) focusing on the most important precomputation step and then verify it in simulated conversation environments (§4.2).

4.1 A Simplified Implementation

Leveraging the instruction-following capability of LLMs, we design prompts to guide them through the key operations of the proactive thinking paradigm, building on the concept of predictive branching.

Formally, at the t-th dialogue turn, immediately after the LLM agent generates its response a_t but before observing the next user utterance o_t, we prompt the model to hypothesize k plausible user replies, denoted $\hat{O}_t = \{\hat{o}_t^{(i)}\}_{i=1}^k$. For each hypothesized reply, the model precomputes a corresponding follow-up action $\hat{a}_{t+1}^{(i)}$ using chain-of-thought reasoning:

$$\hat{O}_t = \pi_\theta(P_h, \tau_{<t}, a_t),$$
$$\hat{a}_{t+1}^{(i)} = \pi_\theta(P_a, \tau_{<t}, a_t, \hat{o}_t^{(i)}),$$

where P_h and P_a are the prompts used for hypothesis generation and action precomputation, respectively. The resulting set of hypothesis and action pairs $\{(\hat{o}_t^{(i)}, \hat{a}_{t+1}^{(i)})\}_{i=1}^k$ constitutes the precomputed rollout set $\mathcal{R}_t$.

We then structure $\mathcal{R}_t$ as a concise, formatted output. This prevents lengthy intermediate reasoning from inadvertently influencing the model in subsequent steps. Determining an optimal number of hypotheses is non-trivial; for experimental verification, we use $k = 3$ by default. We note that this fixed choice is not suitable for real-world deployment but serves specifically to validate the proactive thinking concept.

Finally, upon receiving the actual user reply o_t, the LLM is prompted to produce the next response by conditioning on both the dialogue history and the precomputed results:

$$a_{t+1} = \pi_\theta(P_r, \tau_{<t+1}, \mathcal{R}_t),$$

where P_r is the response generation prompt. For simplicity, we omit cases where precomputation fails and an adaptive thinking process is required. Thus, this implementation could underperform relative to the standard reactive thinking approach.

4.2 Simulated Environments for Verification

Datasets We conduct our experiments on two interactive benchmarks:

- **TelepathyGym** (Qian et al., 2025): This benchmark evaluates an agent's ability to perform strategic reasoning and hypothesis testing through interactive "20 Questions"-style mind-reading games. The core task requires the agent to identify the user's hidden entity by asking strategic yes-or-no questions and making a final guess. The dataset comprises 401 diverse entities spanning categories such as people, animals, fruits, and movies.

- **AgentClinic** (Schmidgall et al., 2024): This benchmark evaluates LLM agents in simulated clinical environments. A doctor agent must diagnose a patient's condition through dialogue to collect information actively. We adopt the dialogue-only setting, AgentClinic-MedQA, which is grounded in 214 cases from the US Medical Licensing Exam (Jin et al., 2021).

We implement the user (for TelepathyGym) and patient (for AgentClinic) roles using DeepSeek-V3.2, mostly following the original study prompts. For AgentClinic, we further refine the patient simulation prompt using an improved version from (Gong et al., 2025) and update the evaluation prompt with stricter rules to prevent ambiguous predictions from being incorrectly scored as correct (see Appendix A for used prompts). Due to the relatively small size of the AgentClinic dataset, we report the average score over 4 runs for stable results.

Evaluation Metrics We evaluate each method for task performance and dialogue latency. Following prior work, task performance is measured by **Accuracy** and **#Turn**. Accuracy indicates whether the agent successfully achieves the goal (i.e., guessing the entity or making the correct diagnosis). #Turn measures interaction efficiency, where a lower number indicates greater efficiency. Evaluating dialogue latency in a simulated environment is challenging because the duration of real-world human interaction between dialogue turns is difficult to estimate, and computational latency depends on hardware. Given pilot study results showing a high correlation between the number of generated tokens and time consumption, we use **#Token** as a proxy for latency, where a lower token count indicates lower latency.

Models and Hyperparameters We conduct experiments using Qwen3-32B (Yang et al., 2025) and DeepSeek-V3.2 (Liu et al., 2025), two recent LLMs demonstrating strong performance across various tasks. For text generation, we consistently use a temperature of 0.5 to mitigate potential degeneration when generating long responses with reasoning traces. We deploy Qwen3-32B using vLLM (Kwon et al., 2023) on four A100 GPUs, while for DeepSeek-V3.2 we utilize the official API service.

Baselines To compare with our **Proactive Thinking** implementation (see §4.1), we include the following baselines:

- **Without Thinking:** The agent generates actions directly without an explicit reasoning process. For TelepathyGym, we directly prompt the LLM to generate its utterance to the user. For AgentClinic, we allow very brief reasoning to ensure responses are well-formatted.

- **Reactive Thinking:** The conventional reasoning baseline where the agent begins thinking *after* receiving the user's response and then produces an action. For Qwen3-32B, instead of using a built-in thinking mode, we prompt it to perform step-by-step reasoning for a fair comparison with our method.

	TelepathyGym		AgentClinic	
	Acc. ↑	#Turn ↓	Acc. ↑	#Turn ↓
Qwen3-32B				
Without	0.681	14.1	0.382	**5.02**
Reactive	0.721	**12.0**	**0.432**	5.55
Proactive	**0.726**	12.3	0.411	5.42
DeepSeek-V3.2				
Without	0.716	13.4	0.498	**5.10**
Reactive	0.855	10.3	0.533	5.21
Proactive	**0.868**	**10.2**	**0.539**	5.27

Table 1: Comparison of without thinking (Without), reactive thinking (Reactive) and proactive thinking (Proactive). We highlight the best result in **bold** and second-best with underline.

The prompts used for these baselines are provided in Appendix A.

4.3 Main Results

Table 1 and Figure 3 show the test results of different methods on TelepathyGym and AgentClinic. We observe that conducting thinking consistently improves task performance on both datasets. For TelepathyGym, thinking not only improves accuracy but also reduces the number of interaction turns, demonstrating higher efficacy. Most significantly, reactive thinking achieves a 0.14-point improvement in accuracy when using DeepSeek-V3.2. For AgentClinic, although dialogue turns increase when using Qwen3-32B, the increase is relatively limited while the accuracy improvement is notable. However, this improvement comes at the cost of increased token consumption, as shown in Figure 3. Users may experience noticeable pauses during the conversation, resulting in a poorer experience.

For proactive thinking, we observe competitive task performance compared to reactive thinking on TelepathyGym. This is expected because user replies are limited to yes, no, or maybe, which are easily predictable. For AgentClinic, we observe a 0.02-point decrease in accuracy compared to reactive thinking with Qwen3-32B. This may be because patient responses are highly flexible, making it difficult for proactive thinking to anticipate all possible replies. Further analysis is provided in §4.4. Additionally, effectively leveraging proactive thinking results for decision-making can be challenging, as existing LLMs may not have been trained on such instructions. This observation is supported by results from DeepSeek-V3.2: with its stronger capabilities, our method successfully maintains task performance. Nevertheless, proactive thinking still consistently outperforms the without-thinking baseline on both models.

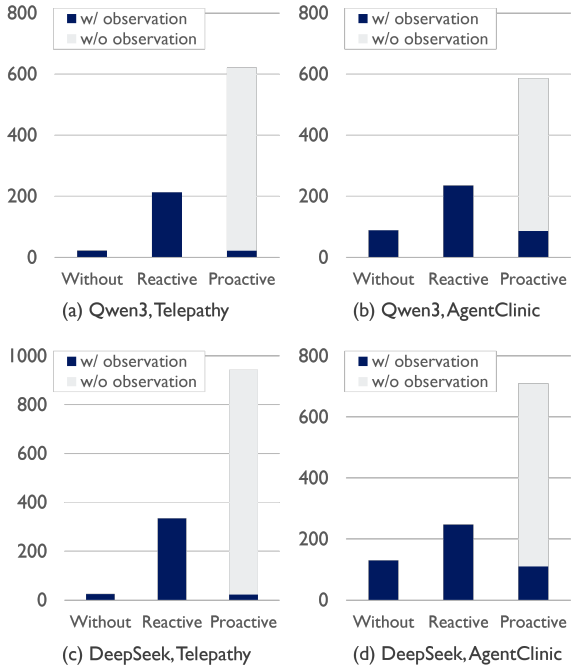

Figure 3: A comparison of token consumption for different methods, where proactive thinking performs precomputation without knowledge of the next user reply (w/o observation).

Results on token consumption show that although proactive thinking consumes many more tokens for reasoning, these can be decoupled from the latest environment observation (i.e., the user reply). After receiving the latest reply, only the same number of tokens as the without-thinking approach is required. This demonstrates the promise of proactive thinking techniques in addressing latency issues.

4.4 Analyses

We further analyze the recall of actual user replies and the number of interaction turns to validate the effectiveness of our proactive-thinking paradigm.

Analyzing the Influence of Parameter k The hyperparameter k plays a critical role in our proactive-thinking implementation. Intuitively, it can increase the recall of the actual user reply by allowing the enumeration of more possible candidates. To understand its impact, we conduct an experiment with different values $k = 1, 2, 3, 5$.

	Acc. ↑	#Turn ↓	Accept ↑
$k = 1$	0.401	5.54	0.640
$k = 2$	0.396	5.44	0.727
$k = 3$	0.411	5.42	0.764
$k = 5$	0.410	5.39	0.772

Table 2: Performance of Qwen3-32B on AgentClinic with different k, where Accept denotes the acceptance ratio of precomputed results.

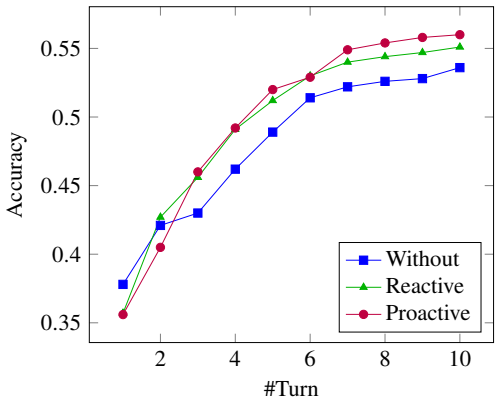

Figure 4: Performance of DeepSeek-V3.2 on AgentClinic at different dialogue turns.

Computing whether the actual user reply is predicted within the top-k predictions is non-trivial due to the diverse expression styles in conversation. Therefore, we calculate the **acceptance ratio**. That is, whether the agent adopts any of the precomputed replies in its next response. This indicates that the corresponding predicted user reply is semantically related to the actual one. Results are shown in Table 2. We observe a consistent improvement in acceptance ratio as k increases, which is expected. This explains why proactive thinking improves performance: the user reply is predictable, and follow-up actions can be prepared in advance. With larger k, we generally observe improved accuracy. This correlation shows that successful recall of the actual user response translates into better task performance. Surprisingly, even with only $k = 1$, the method effectively improves accuracy compared to the without-thinking baseline. This is encouraging because it indicates that **even when the proactive-thinking budget is limited, performance can still be enhanced**, validating the practicality of our approach.

Analysis of Performance Across Dialogue Turns Dialogue turn number is another important aspect of task performance. An expected performance gain should not come at the cost of an increased number of turns. Therefore, we analyze accuracy at different dialogue stages, not only at the final turn. We use the dialogue histories generated by different methods with DeepSeek-V3.2 and consistently employ DeepSeek-V3.2 with a diagnosis-specific prompt (see Figure 12) to perform diagnoses across histories of varying turn lengths. This ensures a fair evaluation among different methods. Results in Figure 4 show that the accuracy of all methods improves as the number of turns increases because more information is gathered. Compared to the results in Table 1, accuracy at the maximum turn is higher for all methods because the evaluation prompt enables deeper reasoning specifically for diagnosis. Except in the early stages, where results may have higher variance, both reactive thinking and proactive thinking consistently outperform the baseline method without thinking. Proactive thinking performs comparably to reactive thinking, demonstrating its effectiveness at each individual dialogue turn.

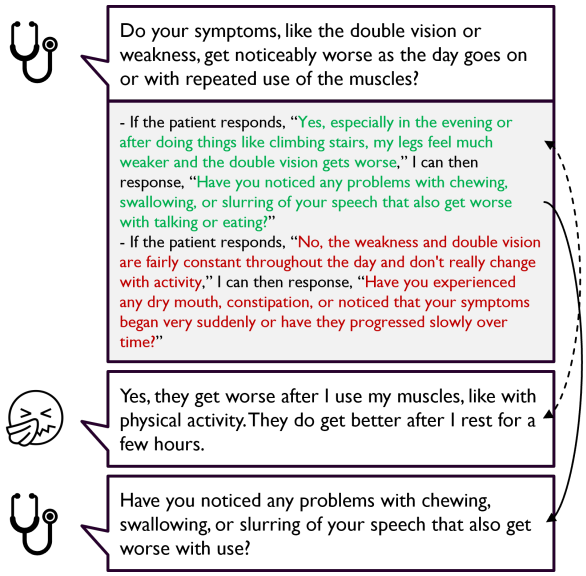

Figure 5: An example from AgentClinic, where the doctor is played by DeepSeek-V3.2.

Case Study on How Proactive Thinking Works Figure 5 presents a case from AgentClinic in which the doctor agent is played by DeepSeek-V3.2. After asking the first question, the agent immediately considers how the patient might reply and drafts corresponding follow-up responses in advance. Once the patient's actual reply is received, the agent compares it with its proactive thinking results and can directly adopt the pre-

pared questions without further reasoning. One possible issue, however, is that the actual patient reply may differ from the most similar anticipated response. For example, the patient adds the statement, "*They do get better after I rest for a few hours*." Ignoring such additional information could compromise performance. This observation suggests the need for adaptive thinking, the ability to judge whether further reasoning is necessary and to refine the next response accordingly.

5 Discussion and Future Directions

5.1 Optimization of Proactive Thinking

While we empirically validate the effectiveness of the proactive thinking paradigm, we acknowledge several critical weaknesses in our current implementation and evaluation, including potential task performance degradation and computational inefficiency. We believe significant room for improvement remains:

- **Predicting Environment State Transitions:** From a predictive branching perspective, the performance of a proactive agent is bounded by its ability to model the prior distribution of the next utterance given the current context. This is conceptually equivalent to predicting state transitions in a Markov decision process. Recent research (Zhang et al., 2025b; Chen et al., 2025) has focused on forecasting the next environment state, notably through world models (Ha and Schmidhuber, 2018; LeCun, 2022; Assran et al., 2025), an area we expect could substantially benefit proactive thinking.

- **Tailored Reasoning:** The pipeline implementation in §4.1 results in obvious computational inefficiency. Since multiple requests share similar context (e.g., the conversation so far), LLMs may perform repetitive reasoning, leading to redundant computation. The fundamental issue is that existing LLMs lack reasoning abilities specifically tailored for proactive thinking. Inspired by the success of DeepSeek-R1 (Guo et al., 2025), we believe this can be addressed through reinforcement learning (Schulman et al., 2017; Shao et al., 2024). Based on the design in §3.1, we can train LLMs to actively explore efficient reasoning strategies that optimize for the final goal, instead of relying on a handcrafted inference framework. However, we note that latency, as an important optimization goal, should be simulated in the training environment. This ensures the agent progresses toward a well-defined objective with minimal bias relative to real-world applications.

Furthermore, the proactive thinking paradigm can be integrated with complementary techniques. For instance, Shih et al. (2025) proposes beginning the thinking process while a speaker is still talking, though this is currently constrained to the speech domain. Similarly, Xie et al. (2025) proposes interleaving thinking with response generation but remains within the reactive thinking paradigm without fully utilizing the conversational structure. Merging proactive thinking with such approaches could conceptually reduce latency or enhance performance.

5.2 From Reactive Loops to Prepared Agents

This work has discussed the feasibility of proactive thinking and validated it through empirical results in dialogues. We hope this paradigm can be extended to more diverse interactive environments, where agents proactively conduct reasoning on demand, thereby decoupling from the reactive cycle. For example, in real-time strategy games (Ma et al., 2024), an agent can continuously simulate an opponent's moves and precompute counter-strategies, enabling instant, high-quality decisions. Similarly, in autonomous driving (Wang et al., 2023), a system can proactively predict the trajectories of nearby vehicles and pedestrians during stable driving periods, allowing for smoother and safer real-time trajectory adjustments. In summary, we believe the shift from reactive loops to prepared agency represents a promising path toward more intelligent interactive AI.

6 Conclusion

In this work, we propose proactive thinking to reduce the latency of response generation in multi-turn dialogues. Unlike standard reactive reasoning, which must wait for the latest observation to begin, our approach decouples the thinking process from the sequential interaction trajectory by precomputing thoughts for later use. Its effectiveness can be intuitively understood from a predictive branching perspective, and we empirically verify it through experiments on two typical tasks. This study serves as a preliminary exploration, and we believe it can enable wider applications and more practical AI systems.

Limitations

As a position paper on the concept of proactive thinking, this study does not fully implement the proposed systems with all necessary capabilities, as doing so would require substantial optimization efforts. A more detailed discussion on this is provided in §5.1. Furthermore, this work validates the effectiveness of proactive thinking only in a controlled simulated environment, which differs from real-world conditions. In particular, determining the appropriate amount of time to allocate for proactive thinking remains a non-trivial challenge. Developing better methods for evaluating proactive thinking continues to be a critical area for future work.

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*.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, and 1 others. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*.

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

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Sara Cao, Yuanhao Xiong, Haibo Tong, and 1 others. 2025. Scaling agent learning via experience synthesis. *arXiv preprint arXiv:2511.03773*.

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*.

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*.

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).

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*.

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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.

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*.

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

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.

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*.

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*.

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*.

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.

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*.

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

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, and 1 others. 2025b. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*.

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*.

A Prompts Used In This Study

This appendix presents the prompts used in this research. As the prompts for TelepathyGym are sourced directly from that framework, we present only the prompts for AgentClinic; for the TelepathyGym prompts, please refer to (Qian et al., 2025).

The prompts for without thinking and reactive thinking are shown in Figure 6 and 7, respectively. For proactive thinking, the prompts for user reply hypothesis and final generation with the actual patient reply are provided in Figure 8 and Figure 9. The follow-up response generation with hypothesized user reply uses the same prompt with reactive thinking.

We employ the patient simulation prompt from (Gong et al., 2025) and show it in Figure 10, which is designed to better evaluate the inquiry capability of a doctor agent. Finally, the prompt for judging the correctness of the predicted diagnosis is shown in Figure 11.

Following (Qian et al., 2025), we instruct LLMs to provide their output summarized in a JSON-like structure. This allows us to easily parse the final dialogue utterance from the full model response.

Doctor Response Generation (Without Thinking)

You are a doctor named Dr. Agent who responds only in dialogue. You are examining a patient and will ask questions about their symptoms, social history, personal information, medical test results, and other relevant details in order to diagnose their condition. The total number of interactions with the patient is limited, so you must balance your questions to reach a diagnosis as efficiently as possible. Your dialogue should be only 1 sentence long. Once you have decided to make a diagnosis, please type "DIAGNOSIS READY: [specific disease name without ambiguity]". Remember, you must identify their disease by asking questions.

Below is all of the information you have:
{basic_information}

Your Response
{
"thought": "Your very brief reasoning about whether to ask a question and what question to ask, or to make a diagnosis based on the information you have.",
"response": "Your question to the patient, 1 sentence long, or DIAGNOSIS READY: [disease name] if you decide to make a diagnosis."
}

Here is the dialogue history:
{dialogue_history}

Now please continue your dialogue

Figure 6: The prompt for doctor response generation when using without-thinking paradigm. We allow a very brief reasoning, which helps generating well-formatted response. The system prompt and user prompt is separated with the solid line.

Doctor Response Generation (Reactive Thinking)

You are a doctor named Dr. Agent who responds only in dialogue. You are examining a patient and will ask questions about their symptoms, social history, personal information, medical test results, and other relevant details in order to diagnose their condition. The total number of interactions with the patient is limited, so you must balance your questions to reach a diagnosis as efficiently as possible. Your dialogue should be only 1 sentence long. Once you have decided to make a diagnosis, please type "DIAGNOSIS READY: [specific disease name without ambiguity]". Remember, you must identify their disease by asking questions.

Below is the information known before the dialogue begins:
{basic_information}

Here is the dialogue history:
{dialogue_history}

Please first write down your step-by-step reasoning process and finally summary using the following format:
Reasoning Process:
... your reasoning process here ...

{
"thought": "Your very brief reasoning about whether to ask a question and what question to ask, or to make a diagnosis based on the information you have.",
"response": "Your question to the patient, 1 sentence long, or DIAGNOSIS READY: [disease name] if you decide to make a diagnosis."
}

Figure 7: The prompt for doctor response generation when using reactive thinking paradigm.

User Reply Hypothesis (Proactive Thinking)

You are a skilled doctor who can anticipate a patient's possible responses.

Below is the information known before the dialogue begins:
{basic_information}

Here is the dialogue history:
{dialogue_history}

Based on this information, please enumerate one most likely plausible response the patient might give. This response should be only one sentence.

Please first write down your step-by-step reasoning process, and then summarize using the following format:
Reasoning Process:
... your reasoning process here ...

{
"patient response": "The most possible response based on the patient's likely description."
}

Figure 8: The prompt for user reply hypothesis when using proactive thinking paradigm.

Doctor Response Generation (Proactive Thinking)

You are a doctor named Dr. Agent who responds only in dialogue. You are examining a patient and will ask questions about their symptoms, social history, personal information, medical test results, and other relevant details in order to diagnose their condition. The total number of interactions with the patient is limited, so you must balance your questions to reach a diagnosis as efficiently as possible. You will be provided with analyses of potential patient responses and corresponding diagnostic considerations. Use this information to guide your questioning. Your dialogue should be only 1 sentence long. Once you have decided to make a diagnosis, please type "DIAGNOSIS READY: [specific disease name without ambiguity]". Remember, you must identify their disease by asking questions.

Below is all of the information you have:
{basic_information}

Please first write down your step-by-step reasoning process and finally summary using the following format:
Reasoning Process:
... your reasoning process here ...

{
"thought": "Your very brief reasoning about whether to ask a question and what question to ask, or to make a diagnosis based on the information you have.",
"response": "Your question to the patient, 1 sentence long, or DIAGNOSIS READY: [disease name] if you decide to make a diagnosis."
}

Here is the dialogue history:
{dialogue_history}

You have previously received the following analytical insights before the patient's latest response:
{precomputation_results}

The patient has now responded as follows:
{actual_patient_response}

Now please continue your dialogue

Figure 9: The prompt for doctor response generation when using proactive thinking paradigm.

Patient Response Generation

You are a patient in a hospital and must answer the doctor's questions based on the context paragraph. Always refer to yourself in the first person unless you are an infant, unconscious, or deceased; in those cases, refer to the patient as your family member. Reveal only the information that is directly asked for. For example, if a question asks generally about your symptoms, provide only your primary symptom, excluding further details such as duration, location, or severity. If the context paragraph contains no relevant information, you must express uncertainty rather than making assumptions.

Your reply must be 1-3 sentences in length. Do not explicitly state your disease, but you may convey other available information in the form of dialogue if asked.

Below is all of your information:
{basic_information}

Your Response
{
"thought": "Your reasoning about how the doctor's question relates to the patient information, revealing only what is directly asked for.",
"response": "Your response to the doctor, 1-3 sentences long."
}

Here is a history of your dialogue:
{dialogue_history}

Now please continue your dialogue

Figure 10: The prompt for patient simulation.

Answer Correctness Judgement

You are responsible for determining whether the current diagnosis and the doctors diagnosis refer unambiguously to the same disease.

Your Response
{
"thought": "Examine whether the doctor's dialogue provides a specific diagnosed disease name, then compare it with the provided correct diagnosis. Note that if the doctor provides an ambiguous diagnosis—such as no specific disease name or multiple possible diseases—it should be considered incorrect.",
"response": "Yes" or "No"
}

Here is the correct diagnosis: {gold_answer}

Here was the doctor dialogue: {last_dialogue_turn}

Did the doctor predict the correct diagnosis without any ambiguity?

Figure 11: The prompt for judging the predicted answer correctness.

Diagnosis Prediction

You are a doctor. Your task is to formulate a differential diagnosis based on the given information and conversation. To make a precise diagnosis, you should analyze the patient's condition, develop a list of possible diagnoses, and finally determine the most likely one.

Below is the information known before the dialogue begins:
{basic_information}

Here is the dialogue history:
{dialogue_history}

Please first write down your step-by-step reasoning process and finally state the most likely diagnosis using the following format:
Reasoning Process:
... your reasoning process here ...

```
{
   "response": "DIAGNOSIS READY: [disease name]"
}
```

Figure 12: The prompt for making a diagnosis only based on the dialogue history.