Exchange-of-Thought: Enhancing Large Language Model Capabilities through Cross-Model Communication

Zhangyue Yin[◇] **Qiushi Sun**[♡] **Cheng Chang**[◇]
Qipeng Guo[◇♣] **Junqi Dai**[◇] **Xuanjing Huang**[◇] **Xipeng Qiu**[◇*]
[◇]School of Computer Science, Fudan University
[♡]National University of Singapore [♣]Shanghai AI Laboratory
{yinzy21,changc21,jqdai22}@m.fudan.edu.cn qiushisun@u.nus.edu
{qpguo16, xjhuang, xpqiu}@fudan.edu.cn

Abstract

Large Language Models (LLMs) have recently made significant strides in complex reasoning tasks through the Chain-of-Thought technique. Despite this progress, their reasoning is often constrained by their intrinsic understanding, lacking external insights. To address this, we propose Exchange-of-Thought (EoT), a novel framework that enables cross-model communication during problem-solving. Drawing inspiration from network topology, EoT integrates four unique communication paradigms: Memory, Report, Relay, and Debate. This paper delves into the communication dynamics and volume associated with each paradigm. To counterbalance the risks of incorrect reasoning chains, we implement a robust confidence evaluation mechanism within these communications. Our experiments across diverse complex reasoning tasks demonstrate that EoT significantly surpasses established baselines, underscoring the value of external insights in enhancing LLM performance. Furthermore, we show that EoT achieves these superior results in a cost-effective manner, marking a promising advancement for efficient and collaborative AI problem-solving.

"Two heads are better than one."
–English Proverb

1 Introduction

Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023) are revolutionizing the field of Natural Language Processing (NLP) by utilizing vast training corpora and huge computational resources (Bai et al., 2022a; Ouyang et al., 2022; Chowdhery et al., 2022; Zhang et al., 2022; Touvron et al., 2023a, *inter alia*). Although LLMs achieve exemplary performance across a wide range of NLP tasks (Wei et al., 2022a; Chung et al., 2022), they consistently struggle to perform well in reasoning tasks, and this limitation cannot be overcome solely by increasing the size of models (Rae et al., 2022; bench authors, 2023).

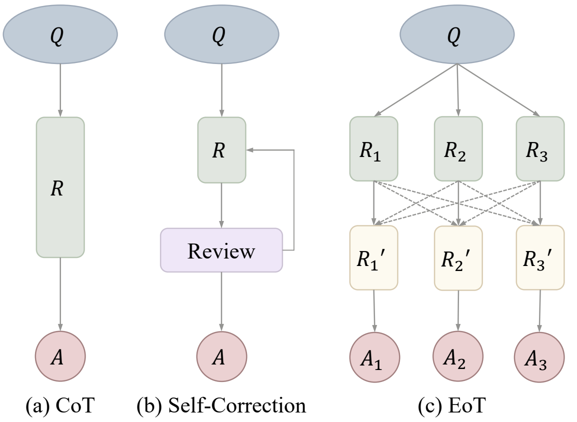


Figure 1: Comparison of CoT, Self-Correction, and EoT. Both CoT and Self-Correction rely on the model's innate abilities to generate and refine output, lacking external insights. EoT enhances the model's reasoning ability by incorporating the thoughts of other models as external insights.

To overcome this shortcoming, Wei et al. (2022b) proposed chain-of-thought (CoT) prompting, which guides the model to generate a series of intermediate reasoning steps before reaching the final answer. At the same time, a series of self-correction methods (Welleck et al., 2023; Ganguli et al., 2023) have been proposed, which aim to iteratively improve the quality of answers by leveraging the model's feedback to their previous outputs (Madaan et al., 2023; Shinn et al., 2023).

However, CoT and self-correction solely base on the model's own understanding and perspective of the question during the reasoning process. Recent studies (Huang et al., 2023; Valmeekam et al., 2023; Stechly et al., 2023) indicate that LLMs struggle to revise their responses without external feedback. This can be attributed to the model's complete dependence on internal representations to generate responses, which makes it difficult to overcome inherent limitations in capability (Yin et al., 2023).

[*] Corresponding author.

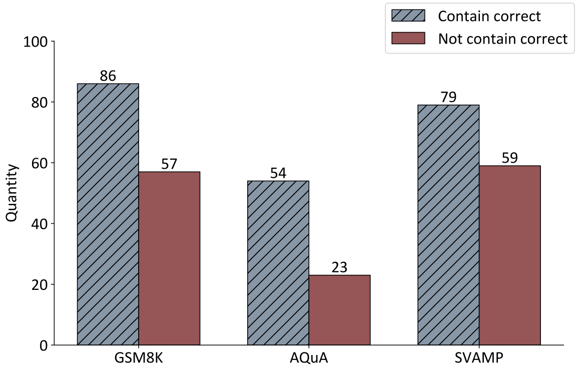


Figure 2: Pilot experiments on three reasoning datasets. The number of erroneous samples containing the correct answer is significantly higher than those not containing the correct answer.

Despite the undeniable importance of external insights (Yao et al., 2023), acquiring high-quality external insights remains a challenge. Wang et al. (2023c)'s research suggests that the single reasoning chain generated by CoT limits the model's reasoning performance. By increasing the temperature to sample diverse reasoning chains and selecting answers through majority voting, the model's reasoning performance can be further improved. However, when confronted with difficult questions, the model often yields a higher number of incorrect responses. In Figure 2, our analysis of correct and incorrect answers within erroneous samples from three reasoning datasets reveals that in most cases the model can deduce the correct answer.

In human society, the truth, even when held by a minority, can gain widespread acceptance and recognition through clear and persuasive communication (Le Bon, 1897). The correct reasoning of others can serve as high-quality external insights, enriching and elevating our collective understanding. Thus, we propose Exchange-of-Thought (EoT), a novel framework that fosters cross-model communication during the problem-solving process. This initiative enables models to incorporate the reasoning of others as external insights.

Figure 1 contrasts EoT with CoT and self-correction methods, highlighting the unique approach of EoT in integrating external perspectives. Inspired by the principles of network topology (Bisht and Singh, 2015) and agent communication (Parsons and McBurney, 2003), we propose four communication paradigms: Memory, Report, Relay, and Debate. These paradigms are designed to facilitate the exchange of ideas and reasoning chains among models, enriching the problem-solving process with a diversity of insights. Furthermore, we delve into the intricacies of each communication paradigm, analyzing the dynamics of information flow and the volume of communication. With the awareness that both correct and incorrect reasoning chains propagate within communications, we introduce confidence evaluation mechanisms that employs the analysis of answer variations to assess models' confidence levels. It is designed to mitigate the influence of erroneous reasoning, thereby ensuring the integrity and reliability of the problem-solving process.

Experiments across various complex reasoning tasks demonstrate that EoT significantly outperforms established strong baselines, underscoring the critical role of external insights in augmenting the capabilities of LLMs. We summarize our contributions as follows:

- We introduce Exchange-of-Thought (EoT), a pioneering framework for cross-model communication that incorporates external insights from other LLMs during problem-solving.
- We present and examine four communication paradigms coupled with a confidence evaluation mechanism that assesses model certainty through the variability of answers, mitigating the impact of incorrect reasoning.
- Experimental results on various complex reasoning tasks underscore the efficacy and cost-effectiveness of EoT, highlighting the significance of incorporating external insights and communication in problem-solving.

2 Related Work

2.1 Chain-of-Thought prompting in LLMs

Wei et al. (2022b) highlight that LLMs can manifest enhanced reasoning capabilities when being prompted by demonstrations with intermediate reasoning steps. This technique can effectively improve the performance of LLMs on complex reasoning tasks (Wei et al., 2022a; Kojima et al., 2022). A series of strategies for enhancing CoT has been proposed to further improve the performance of LLMs. One such method is program-aided language models (Gao et al., 2022; Chen et al., 2022), which aims to decouple reasoning and computation through program synthesis. Moreover, complex tasks can also be transformed into delegable sub-tasks through modular approaches (Khot et al., 2023). Choosing appropriate demonstrations can also enhance the performance of CoT (Li et al., 2023a; Li and Qiu, 2023a). Notable among these, AutoCoT (Zhang et al., 2023b) uses an automated way to construct and sample diverse demonstrations. Active-Prompt (Diao et al., 2023) selects the most helpful samples for labeling based on the model's uncertainty in the outputs. Recently, Li and Qiu (2023b) employ a strategy of storing high-confidence thoughts as external memory and retrieves these insights to aid the reasoning process.

2.2 Ensemble of Reasoning Paths

LLMs have the ability to explore multiple reasoning paths using techniques such as temperature adjustment and prompt sampling (Chu et al., 2023). Wang et al. (2023c) suggest that for complex questions, there may be several correct paths to approach a problem, leading to the proposal of Self-Consistency. This method replaces the greedy decoding strategy with the sampling of multiple reasoning paths and selecting the most consistent answer, resulting in significant performance improvements. Beyond that, Fu et al. (2023b) discover that prompts with higher reasoning complexity could achieve better performance in multi-step reasoning tasks, leading to the proposal of complexity-based prompting. While other methods, such as re-ranking (Cobbe et al., 2021; Thoppilan et al., 2022), have also been applied to select suitable reasoning paths, they often rely on heuristic or trained smaller models. Recently, Li et al. (2023b) sample different demonstrations and use step-by-step verification to filter out incorrect answers. However, obtaining step-level labels can be challenging, and using smaller models for judgment struggles to handle complex reasoning processes. In contrast, our method fully utilizes the communication and decision-making capabilities of LLMs to reach the final answer, without the need for additional training and annotated data.

2.3 Reasoning Path Refinement

Although CoT (Wei et al., 2022b) effectively enhances the performance of LLMs in complex reasoning tasks, they remain susceptible to errors during the reasoning process, leading to incorrect answers (Bai et al., 2022b; Lyu et al., 2023). To mitigate this issue, starting from the model's own thoughts, Shinn et al. (2023) and Madaan et al. (2023) employ the model's own feedbacks and past mistakes to refine the reasoning process. Yao et al. (2023) explore the synergies between reasoning chains and action plans. For numerical problems, Zheng et al. (2023) gradually guide models to the correct answer by using previously generated answers as hints. With the aid of external knowledge, Wang et al. (2023a) introduce chain-of-knowledge prompting that employs evidence triples to curb the generation of unfactual and unfaithful answers. Taking model interactions into account, multi-agent debates (Du et al., 2023; Liang et al., 2023) have been introduced to enhance the factual accuracy of generated content and reduce fallacies and hallucinations. EoT differs from these efforts as we prioritize enhancing the current reasoning process generated by a single model by incorporating the reasoning processes from other models as external insights through cross-model communication.

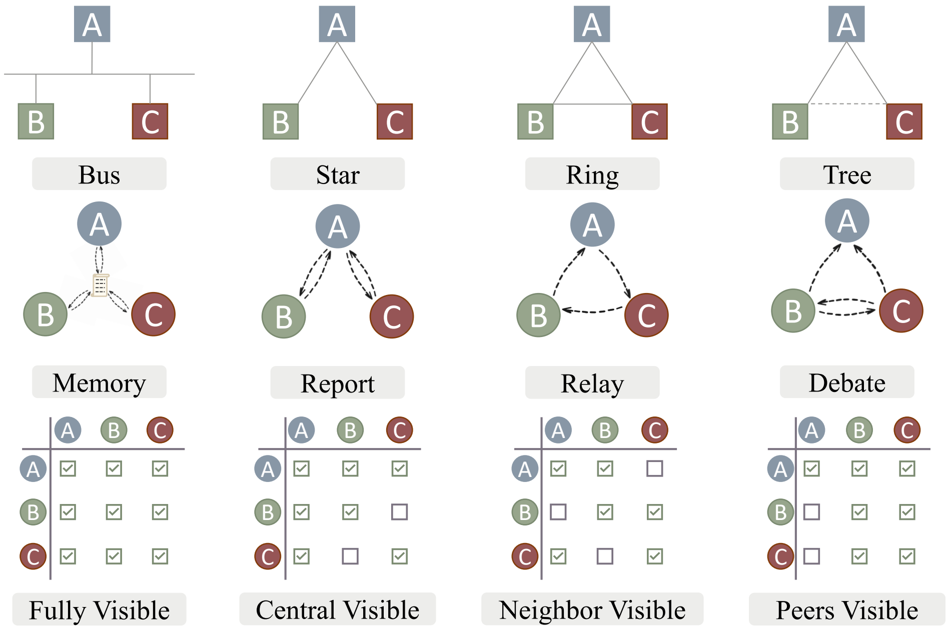
3 Preliminary

Firstly, we define the current methods that use LLMs to solve problems. We denote a LLM with a parameter size of θ as p_θ, and the sequence length as t, which includes tokens $[s_1, s_2, \dots, s_t]$. The LLM predicts the next token based on the prior tokens in the sequence. The probability of the s_i token is $p_\theta(s_i|s_1, s_2, \dots, s_{i-1})$. Therefore, the probability of the whole sentence is $p_\theta(s) = \prod_{i=1}^{t} p_\theta(s_i|s_{\leq i-1})$.

Standard prompting. Standard prompting involves deriving an answer a from a question q using $p_\theta(a|q)$. In-Context Learning (Brown et al., 2020) aims to improve LLMs performance by adding demonstrations $D = \{d_1, d_2, \dots, d_n\}$ to the input, which can be expressed as $p_\theta(a|D, q)$.

CoT prompting. As identified by Wei et al. (2022b), the incorporation of intermediate reasoning steps can improve the proficiency of LLMs in tackling complex reasoning challenges. To facilitate this, a rationale r_i is added to demonstration $d_i = \{q_i, r_i, a_i\}$ to guide the LLMs in explicitly generating reasoning steps. Fu et al. (2023b) observe that using rationale r_i with more complex reasoning steps for demonstrations can further enhance the model's reasoning performance.

Self-Consistency. Self-Consistency method, introduced by Wang et al. (2023c), effectively consolidates answers from multiple independent reasoning chains. This technique prioritizes the most commonly occurring answer, defined as $a = \operatorname{argmax}_{a_i} f(a_i)$, where $f(a_i)$ denotes the frequency of each answer a_i. This approach enables



Fully Visible

	A	B	C
A	☑	☑	☑
B	☑	☑	☑
C	☑	☑	☑

Central Visible

	A	B	C
A	☑	☑	☑
B	☑	☑	□
C	☑	□	☑

Neighbor Visible

	A	B	C
A	☑	☑	□
B	□	☑	☑
C	☑	□	☑

Peers Visible

	A	B	C
A	☑	☑	☑
B	□	☑	☑
C	□	☑	☑

Figure 3: Correspondence between communication paradigms and network topologies. The top row depicts four network topologies. The second row correlates these with the corresponding communication paradigms. The bottom row offers an analysis of the communication volume associated with each paradigm. The horizontal axis represents the information that the node can receive, while the vertical axis indicates the information that the node can send.

the model to explore a broader range of reasoning pathways, thereby enhancing its reasoning ability. However, it remains constrained by the intrinsic limitations of LLMs' capabilities.

Progressive-Hint Prompting. Introduced by Zheng et al. (2023), Progressive-Hint Prompting (PHP) leverages a sequence of historical answers $\{a^{(1)}, a^{(2)}, \ldots, a^{(j-1)}\}$ to enhance the current reasoning process $r^{(j)}$ and facilitate the derivation of the subsequent answer $a^{(j)}$.

4 Methodology

We introduce Exchange-of-Thought (EoT), a novel framework designed to facilitate cross-model communication, allowing for the exchange of reasoning processes to integrate external insights. This innovative approach leverages the communicative abilities of LLMs to promote the sharing of rationale r and answer a among participating models, fostering a collaborative environment for thought and analysis. The implementation of EoT encounters three key challenges:

1. How to identify the appropriate counterparts for model communication?
2. What are the conditions for ceasing communication between models?
3. How to minimize the influence of incorrect reasoning during the communication process?

4.1 Communication Paradigm

Inspired by network topology (Bisht and Singh, 2015) and intelligent agent communication (Parsons and McBurney, 2003), we propose four communication paradigms to determine the counterparts for model communication. As illustrated in Figure 3, we propose Memory, Report, Relay, and Debate communication paradigms each corresponding to the Bus, Star, Ring, and Tree network topologies, respectively. Assume in j-th round of communication, given a set of LLMs $\{M\} = \{m_1, m_2, \ldots, m_n\}$, the model m_i generates the corresponding rationale $r_i^{(j)}$ and the answer $a_i^{(j)}$ based on the $(r_K^{(j-1)}, a_K^{(j-1)})$, where K is the set from which model m_i can receive reasoning processes. In the first round, we use the CoT method proposed by Wei et al. (2022b) to generate $(r^{(1)}, a^{(1)}) \sim P_\theta(r^{(1)}, a^{(1)}|D, q)$.

Memory. Under the Memory paradigm, all models record their rationale r and answer a in a logbook, which is fully visible from all models. This means that in the j-th round, any model, such as model m_A, can access the reasoning chains and answers from all models $(r_m^{(j-1)}, a_m^{(j-1)})$, $m \in \{M\}$. As depicted in Figure 3, this paradigm facilitates the fastest flow of information and also incurs the highest communication cost among all paradigms.

Report. Under the Report paradigm, we designate model m_A as the central node, which can obtain the rationale and answer from all other models $(r_m^{(j-1)}, a_m^{(j-1)})$, $m \in \{M\} \backslash \{m_A\}$. Both m_B and m_C only receive information from m_A and do not interact with each other. Consequently, m_A plays a pivotal role in the communication process. This paradigm also allows for rapid information flow, but it demands a higher capacity for processing and analysis for the central node.

Relay. Under the Relay paradigm, we order the models by number and connect them in a circle. Each node is capable of receiving information from the preceding node and transmitting its own information to the subsequent node. For example, in the j-th round, m_A passes $(r_A^{(j-1)}, a_A^{(j-1)})$ to m_C and receives $(r_B^{(j-1)}, a_B^{(j-1)})$ from the previous round of m_B. This distributed communication mode can reduce the demands on the information processing capacity of each node, but it may result in a slower flow of information.

Debate. We have adapted the tree topology to devise the Debate paradigm. This paradigm permits leaf nodes to exchange information with each other, while parent nodes are solely responsible for aggregating information, meaning that information flow is directed upward from child to parent. As illustrated in Figure 3, m_B and m_C, as child nodes, are able to communicate, whereas m_A, as a parent node, can only receive information from its children. This communication paradigm strikes a balance between the model's information processing capacity and the speed of information flow.

4.2 Communication Volume

The last row of figure 3 displays the information that can be transmitted and received in different communication paradigms. The communication volume is measured by the number of messages received, assuming there are n models participating in the communication, with each node transmitting its information from the previous round to the next.

In the Memory paradigm, every node receives information from all other nodes in the previous round, resulting in a communication volume of n^2. Any piece of information requires only one transmission to reach the corresponding node.

Under the Report paradigm, the central node receives information from $n-1$ non-central nodes, while each of the $n-1$ non-central nodes receives information from the central node. In addition, each node can receive information from its previous round. Thus, the total communication volume is $3n-2$. The transmission from a non-central node to another non-central node requires two transmissions, whereas sending to the central node requires only one. Thus, the average communication volume is calculated as $2 - \frac{2}{n}$.

Under the Relay paradigm, each node receives information from the preceding node and its own information from the last round, resulting in a communication volume of $2n$. Node i sends information to node $i+1$ in just one transmission, but sending to node $i-1$ requires $n-1$ transmissions. Therefore, the average propagation speed is $\frac{n}{2}$.

In the Debate paradigm, nodes are assumed to form a full binary tree with a height of $h = \lceil \log_2(n+1) \rceil$. The communication volume for each pair of child nodes is 4, and it is 3 for the parent node. Consequently, a subtree comprising two children and one parent has a communication volume of 7. The number of non-leaf nodes in a full binary tree is $\frac{n-1}{2}$, leading to a total communication volume of $\frac{7(n-1)}{2}$. Information under the same parent node requires only one transmission, whereas the information from the farthest nodes needs $h-1$ transmissions to converge at the root node. Thus, the communication speed $\mathcal{S} = \frac{\Sigma_{i=1}^{h-1} 2^{i-1} i}{2^{h-1}-1}$.

4.3 Termination Condition

Utilizing the models' current round outputs and the answers from previous rounds, we have devised two criteria for terminating communication: consistent output and majority consensus.

Consistent Output Termination. Inspired by Zheng et al. (2023), we implement a consistent output termination in EoT. The termination condition is triggered when the output of model m_i in the j-th round is the same as the output in the $j-1$-th round, $a_i^{(j)} = a_i^{(j-1)}$. In this case, m_i will stop receiving or sending information and exit the current communication.

Majority Consensus Termination. Du et al. (2023) observed that LLMs can converge on a consensus after several rounds of debate, suggesting that LLMs fine-tuned with reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) are more likely to reach an agreement. Inspired by this finding, we propose the termination condition of majority rule, where LLMs cease com-

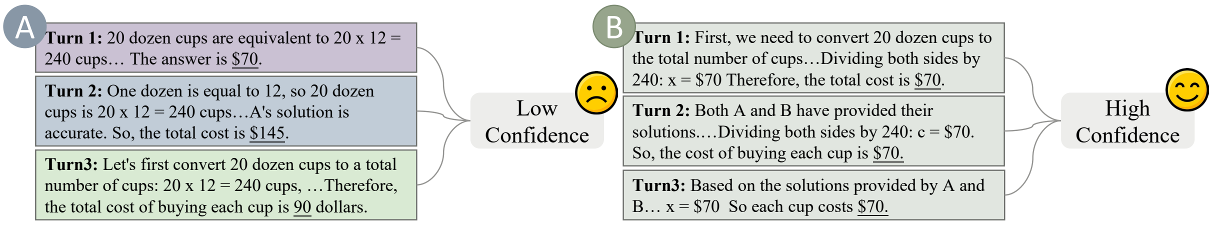

A
Turn 1: 20 dozen cups are equivalent to 20 x 12 = 240 cups… The answer is $70.
Turn 2: One dozen is equal to 12, so 20 dozen cups is 20 x 12 = 240 cups…A's solution is accurate. So, the total cost is $145.
Turn3: Let's first convert 20 dozen cups to a total number of cups: 20 x 12 = 240 cups, …Therefore, the total cost of buying each cup is 90 dollars.
Low Confidence
B
Turn 1: First, we need to convert 20 dozen cups to the total number of cups…Dividing both sides by 240: x = $70 Therefore, the total cost is $70.
Turn 2: Both A and B have provided their solutions.…Dividing both sides by 240: c = $70. So, the cost of buying each cup is $70.
Turn3: Based on the solutions provided by A and B… x = $70 So each cup costs $70.
High Confidence

Figure 4: An illustrative comparison between a confident model and an unconfident model. Model A generates three different answers over three communication rounds, indicating uncertainty about the answer, while Model B consistently adheres to a single answer.

munication with each other once a majority of them reach an agreement. This approach serves as a global termination condition, distinguishing it from the consistent output termination, which acts as a cessation criterion on an individual model basis.

4.4 Confidence Evaluation

An intriguing aspect of human behavior is that individuals are less likely to make mistakes when they are confident in their answers. Conversely, when uncertain about their answers, they become more susceptible to the influence of others' opinions. Additionally, as found by Wang et al. (2023c), the likelihood of an answer being correct decreases as the generated results become more contradictory. Therefore, if a model's answers frequently change during communication, there is a high probability that these answers are incorrect.

We propose calculating the model's confidence based on the variation in responses. This aids the recipient of the information in verifying its reliability, thereby safeguarding the problem-solving process from the disruption of erroneous information. Figure 4 presents an illustrative example of a confident model and a non-confident model.

In a communication with k rounds, model m_i generates a set of answers $\{a_i^{(1)}, \dots, a_i^{(k)}\}$. Let $f(a_i) = \max \#\{a \mid a = a_i^{(j)}\}$ denote the number of the most frequently occurring answer from model m_i. Consequently, we obtain the model's confidence level $\mathcal{C}_i = \frac{f(a_i)}{k}$ in the current round.

5 Experiments

5.1 Experimental Setups

Tasks and Datasets. In our experiments, we evaluated the performance of EoT across three complex reasoning tasks: **(1) Mathematical Reasoning**: This involves six datasets, which includes GSM8K (Cobbe et al., 2021), MultiArith (Roy and Roth, 2015), SingleEQ (Koncel-Kedziorski et al., 2015), AddSub (Hosseini et al., 2014), AQuA (Ling et al., 2017), and SVAMP (Patel et al., 2021). **(2) Commonsense Reasoning**: We utilize the CommonsenseQA(CSQA; Talmor et al., 2019) and StrategyQA (Geva et al., 2021). **(3) Symbolic Reasoning**: We employ two datasets from Big-Bench (bench authors, 2023; Suzgun et al., 2023), namely Penguins in a Table (Penguins) and Date Understanding. In Appendix B, we provide a detailed description and statistics of the datasets.

Baselines. We compare EoT with a series of strong baselines, which include (1) Chain-of-Thought prompting (CoT; Wei et al., 2022b), (2) Complexity-based prompting (ComplexCoT; Fu et al., 2023b), (3) Self-Consistency (SC; Wang et al., 2023c), (4) Progressive Hint Prompting (PHP; Zheng et al., 2023). Specifically, CoT and ComplexCoT are prompting methods, while SC and PHP are reasoning chain ensemble methods. For simplicity in notation, we use "CoT-SC(10)" to denote the approach that employs the CoT prompt method to sample 10 reasoning chains and then utilize the SC method to select the answer.

Implementation Details. We access the GPT models through the OpenAI API. In the main experiments, we employ `GPT-3.5-Turbo-0301` (`GPT-3.5`) and `GPT-4-0314` (`GPT-4`) to evaluate the effectiveness of EoT in comparison to other strong baselines. We set the temperature at 1 during the generation. The prompts for CoT and ComplexCoT are sourced from Wei et al. (2022b) and Fu et al. (2023b). By default, we employ three `GPT-3.5-Turbo-0301` to engage in the EoT communication. We apply the majority consensus termination and confidence evaluation, selecting the majority answer as the final outcome. Taking into account the impact of temperature, we report the average performance and standard deviation across five runs. Additionally, in Section 5.3, to further

Method	GSM8K	MultiArith	SingleEQ	AddSub	AQuA	SVAMP	Avg.
			Single Reasoning Chain				
CoT	79.12±0.50	97.27±0.65	92.80±0.27	86.23±0.52	55.12±1.03	79.52±0.81	81.67
ComplexCoT	79.32±0.65	95.40±0.50	91.34±0.33	84.46±0.86	56.46±0.59	77.70±0.54	80.78
CoT (GPT-4)	**94.90**	97.80	93.10	**89.30**	**77.50**	**90.50**	90.51
			Ensemble Methods				
CoT-SC(3)	82.82±0.32	98.20±0.43	93.31±0.12	87.19±0.47	62.13±1.30	81.98±0.49	84.27
CoT-SC(5)	85.47±0.52	98.60±0.08	93.70±0.25	87.49±0.38	64.02±0.95	83.76±0.81	85.50
CoT-SC(10)	87.57±0.27	98.97±0.12	94.06±0.36	87.59±0.58	66.38±1.72	84.96±0.33	86.59
ComplexCoT-SC(3)	84.17±0.67	97.43±0.31	92.95±0.53	86.13±0.74	60.47±1.55	81.44±0.79	83.77
ComplexCoT-SC(5)	87.26±0.33	98.13±0.22	94.02±0.29	86.48±0.61	62.05±2.40	83.86±0.92	85.30
ComplexCoT-SC(10)	89.23±0.31	98.23±0.37	94.21±0.16	86.58±0.58	64.96±1.91	85.58±0.87	86.46
PHP	85.10	98.00	92.90	85.30	60.60	83.10	84.16
			Exchange-of-Thought				
EoT-Memory	88.98±0.89	98.80±0.16	94.09±0.48	87.65±0.49	69.37±2.77	84.28±0.48	87.20
EoT-Report	88.61±0.83	**99.03±0.22**	94.06±0.47	87.95±0.34	70.31±2.19	84.78±0.75	87.46
EoT-Relay	88.42±0.72	98.97±0.16	94.13±0.49	87.59±0.58	70.87±1.98	85.04±0.31	87.50
EoT-Debate	88.52±0.76	98.90±0.17	**94.25±0.19**	87.70±0.34	69.69±1.24	85.10±0.24	87.36

Table 1: Comparison of EoT performance with a series of strong baselines on mathematical reasoning tasks. The best results are highlighted in bold, while the best results among different EoT paradigms are underlined. The performance of different EoT communication paradigms is represented by varying colors, with darker shades indicating higher performance. The results for CoT (GPT-4) and PHP are reported from Zheng et al. (2023).

validate the performance of different LLMs on EoT, we incorporate the `Claude-2` model. The further implementation details are listed in Appendix C.

5.2 Performance of EoT

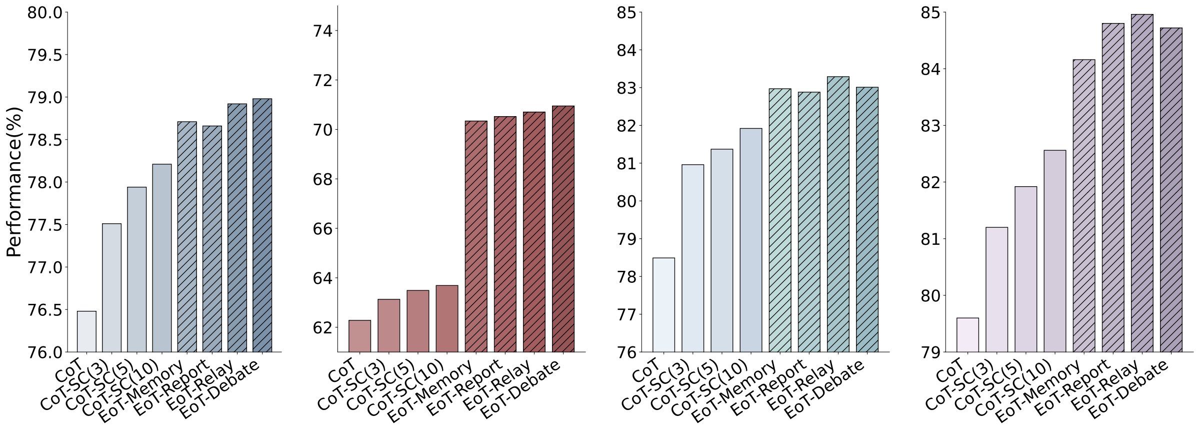
Mathematical Reasoning. According to the results presented in Table 1, the four communication paradigms of EoT have shown significant improvement over both CoT and ComplexCoT in mathematical reasoning tasks. Compared to the currently strongest baseline method, PHP, the Memory, Report, Relay, and Debate paradigms have respectively increased the average performance by 3.04%, 3.30%, 3.34%, and 3.20%. EoT comprehensively outperforms CoT-SC(5), achieving performance comparable to, or even surpassing, that of CoT-SC(10). When compared to the current best LLM `GPT-4`, three `GPT-3.5` with EoT surpassed a single `GPT-4` with CoT on the MultiArith and SingleEQ datasets. This indicates that through cross-model communication and collaboration, three less capable models can compensate for their individual weaknesses and outperform more powerful model, showcasing the potential of EoT to enhance model capabilities and address inherent shortcomings by incorporating external insights.

Commonsense Reasoning. The comparison of EoT with CoT and CoT-SC methods on commonsense reasoning tasks is illustrated in Figures 5a and 5b. EoT significantly outperforms CoT. Specifically, on the StrategyQA dataset, Memory, Report, Relay, and Debate respectively achieve improvements of 8.06%, 8.24%, 8.42%, and 8.67% compared to CoT. Similar significant gains are observed on the CSQA dataset. Furthermore, across both commonsense reasoning tasks, all four paradigms outperform the CoT-SC(10) method, which samples 10 reasoning chains, demonstrating the superior performance of EoT.

Symbolic Reasoning. Figures 5c and 5d compare the performance of EoT with CoT and CoT-SC methods on symbolic reasoning tasks. On the Penguins dataset, the Memory, Report, Relay, and Debate paradigms of EoT achieve improvements of 2.01%, 1.92%, 2.33%, and 2.05% respectively, compared to the CoT-SC(3) method which samples 3 reasoning chains. On the Date Understanding dataset, the performance gains of EoT are even more pronounced, with all four paradigms showing an average improvement of 2.1% over CoT-SC(10).

5.3 Discussions

Communication Paradigm. We propose four communication paradigms and analyze the communication volumes in Section 4.1 and Section 4.2. In the results illustrated in Table 1, we observe that



(a) CSQA.

(b) StrategyQA.

(c) Peguins.

(d) Date Understanding.

Figure 5: Comparison of EoT with CoT and CoT-SC methods in commonsense and symbolic reasoning tasks.

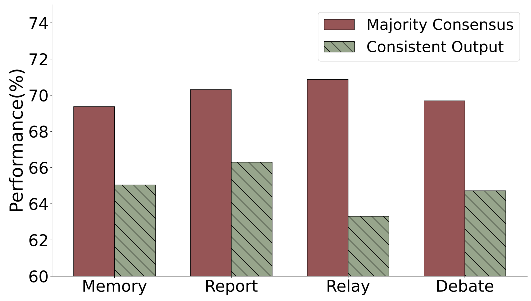


Figure 6: Comparison of consistent output termination and majority consensus termination on AQuA.

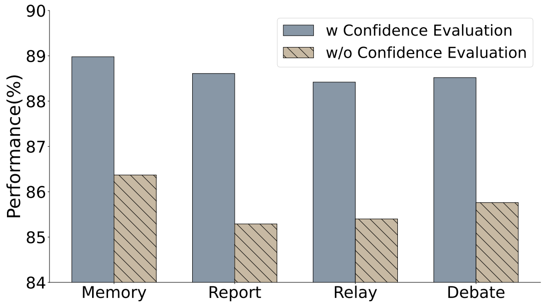


Figure 7: The impact of employing confidence evaluation on accuracy in the GSM8K dataset.

different communication paradigms have their respective strengths. For instance, Report performs best on MultiArith and AddSub, while Debate achieves optimal performance on SingleEQ and SVAMP. This indicates that various communication paradigms are well-suited for different scenarios.

Termination Condition. In Figure 6, we analyze the performance of the two termination conditions we propose in Section 4.3 on the AQuA dataset. Compared to consistent output termination, majority consensus termination improved by 4.33%, 4.01%, 7.56%, and 4.97% under the Memory, Report, Relay, and Debate paradigms, respectively. Under consistent output termination, there is no mechanism for collective negotiation, and individual models are prone to premature exit due to degeneration (Su et al., 2022). Therefore, majority consensus termination is more suitable for scenarios involving multiple model communication.

Confidence Evaluation. We conduct ablation experiments on the GSM8K dataset for confidence evaluation. As shown in Figure 7, across four communication paradigms, confidence evaluation show an average improvement of 2.92% compared to the baseline. The introduction of confidence evaluation enables the model to consider the other model's confidence prior (Zhang et al., 2023a) during communication, facilitating its decision to accept the other model's reasoning chains at an earlier stage, thereby effectively mitigating the interference of incorrect reasoning chains.

Round Analysis. As illustrated in Figure 8, we analyze the number of communication rounds to satisfy termination condition in the SVAMP dataset. For the majority of samples, consensus on the answer can be reached within three rounds of communication. Wang et al. (2023c) obverse that answer consistency is proportional to accuracy. EoT enables models to engage in a greater number of exchanges and discussions on questions where consensus is challenging to achieve. Consequently, a minority of difficult cases necessitate communication extending beyond five rounds.

Cost Analysis. A potential concern is the computational expense incurred by EoT. In Figure 9, we compare the performance and computational costs of CoT-SC, ComplexCoT-SC, and EoT methods. Compared to CoT-SC(5), EoT reduces costs by 20% while enhancing performance by 3%. EoT achieves comparable performance to ComplexCoT-

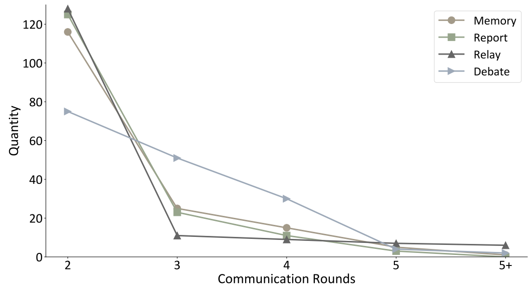


Figure 8: Number of communication rounds required to reach termination condition on SVAMP.

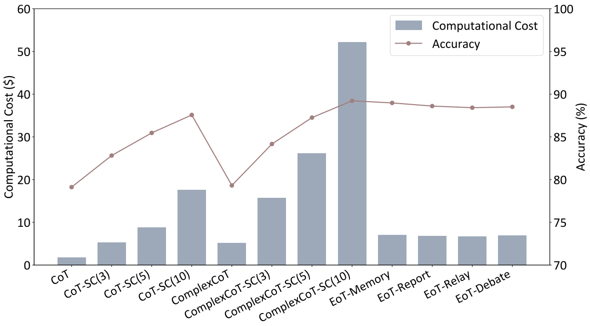


Figure 9: Performance and associated costs of different methods in the GSM8K dataset.

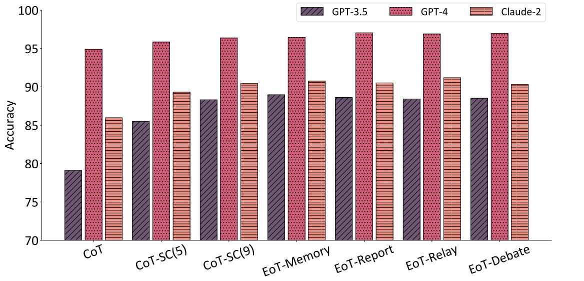


Figure 10: Comparison of EoT with CoT and CoT-SC methods using different LLMs as backbones on GSM8K.

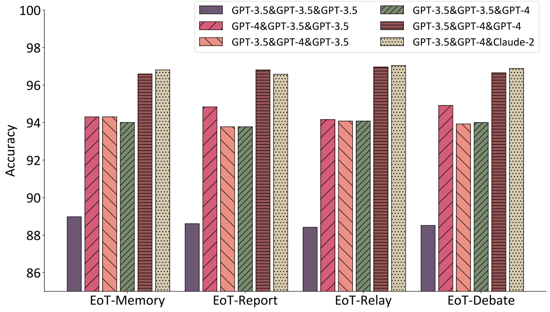


Figure 11: Effect of different node positions for LLMs on accuracy in the GSM8K Dataset.

SC(10) at only one-seventh of its cost. Since the majority of samples conclude communication within three rounds, EoT does not impose a significant computational burden. By facilitating the exchange of external insights between models, EoT effectively enhances model performance, demonstrating a cost-effective advantage.

Model Applicability. In Figure 10, we analyze the performance of EoT when applied to different LLMs. EoT, compared to CoT-SC(5), shows performance improvements of 3.2% on `GPT-3.5`, 1.0% on `GPT-4`, and 1.4% on `Claude-2`, indicating that EoT is adaptable to various LLMs and effectively boosts performance across multiple LLMs.

Position Analysis. In Figure 11, we investigate the impact of different LLMs occupying different node positions on performance. Notably, positioning the more powerful `GPT-4` as the central node in the Report paradigm yields a performance increase of over 1% compared to when `GPT-4` serves as a non-central node. In the Debate paradigm, `GPT-4` as a parent node outperforms `GPT-4` as a child node by 0.9%. The location of `GPT-4` has a negligible effect on the decentralized Relay and Memory paradigms. Additionally, a configuration with two `GPT-4` models and one `GPT-3.5` significantly outperforms one with two `GPT-3.5` models and one `GPT-4`, underscoring that incorporating more superior models can further enhance EoT's performance. The combination of `GPT-3.5`, `GPT-4`, and `Claude-2` achieves performance close to or exceeding that of two `GPT-4` with one `GPT-3.5`, suggesting that model diversity can effectively boost EoT's effectiveness, aligning with the ensemble theory (Kuncheva and Whitaker, 2003) that diversity among models can improve performance.

6 Conclusion

We introduce Exchange-of-Thought (EoT), a novel framework that enriches models with external insights through cross-model communication. We develop four communication paradigms and conduct a thorough analysis of the communication volume and information propagation speed. To safeguard against the disruption of incorrect reasoning processes, we design a confidence evaluation mechanism. Experiment on mathematical, commonsense, and symbolic reasoning tasks demonstrates that EoT surpasses a series of strong baselines while also offering a cost advantage. Further analysis reveals that EoT is adaptable to various models, and the participation of a more diverse range of models can further enhance the performance of EoT.

Ethics Statement

The EoT method presented in this paper does not require the collection or utilization of any personal information. The prompts we have designed and employed are free from personal data and avoid language that discriminates against individuals or groups. We have conducted a comprehensive research of the licenses for the datasets used in this paper, as detailed in Appendix B, and have ensured that our work complies with all the licensing requirements of these datasets.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (No.2022CSJGG0801), National Natural Science Foundation of China (No.62022027). We extend our sincerest gratitude to the reviewers for their insightful comments and suggestions, which have been instrumental in enhancing the quality of this manuscript.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

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

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Nivedita Bisht and Sapna Singh. 2015. Analytical study of different network topologies. *International Research Journal of Engineering and Technology (IRJET)*, 2(01):88–90.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv preprint*, abs/2211.12588.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

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

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *ArXiv preprint*, abs/2302.12246.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *ArXiv preprint*, abs/2305.14325.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023a. Improving language model negotiation with self-play and in-context learning from ai feedback.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023b. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *ArXiv preprint*, abs/2211.10435.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

David Ha and Yujin Tang. 2022. Collective intelligence for deep learning: A survey of recent developments. *Collective Intelligence*, 1(1):26339137221114874.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Ludmila I Kuncheva and Christopher J Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51:181–207.

Gustave Le Bon. 1897. *The crowd: A study of the popular mind*. TF Unwin.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023a. Unified demonstration retriever for in-context learning. *ArXiv preprint*, abs/2305.04320.

Xiaonan Li and Xipeng Qiu. 2023a. Finding supporting examples for in-context learning. *ArXiv preprint*, abs/2302.13539.

Xiaonan Li and Xipeng Qiu. 2023b. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts. *ArXiv preprint*, abs/2305.05181.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *ArXiv preprint*, abs/2305.19118.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics.

Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. 2023. Scaling laws of rope-based extrapolation.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *ArXiv preprint*, abs/2301.13379.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv preprint*, abs/2303.17651.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

S. Parsons and Peter McBurney. 2003. Argumentation-based communication between agents. In *Communication in Multiagent Systems*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Pragaash Ponnusamy, Alireza Ghias, Yi Yi, Benjamin Yao, Chenlei Guo, and Ruhi Sarikaya. 2022. Feedback-based self-learning in large-scale conversational ai agents. *AI magazine*, 42(4):43–56.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2022. Scaling language models: Methods, analysis & insights from training gopher.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1743–1752. The Association for Computational Linguistics.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *ArXiv preprint*, abs/2303.11366.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *ArXiv preprint*, abs/2201.08239.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans?

Aimee Van Wynsberghe. 2021. Sustainable ai: Ai for sustainability and the sustainability of ai. *AI and Ethics*, 1(3):213–218.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. 2023a. Boosting language models reasoning with chain-of-knowledge prompting. *ArXiv preprint*, abs/2306.06427.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023b. Augmenting language models with long-term memory.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E. Gonzalez. 2023a. The wisdom of hindsight makes language models better instruction followers. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *ArXiv preprint*, abs/2304.09797.

A Limitations and Broader Impacts

Given the current constraints in communication and analytical capacities of open-source models (Fu et al., 2023a), as well as their substantial computational resource requirements (Touvron et al., 2023b; Chowdhery et al., 2022), we have not included these models in our experiment at this stage. However, we posit that open-source models with advanced comprehension and communication skills have the potential to match or even exceed the performance of commercial models (OpenAI, 2023; Ouyang et al., 2022; Chowdhery et al., 2022), through the collaborative exchange of insights.

A critical factor in model communication is the handling of long text. The current context windows of these models limit our ability to incorporate a broader range of models in the communication process. Recent works (Liu et al., 2023; Xiao et al., 2023; Wang et al., 2023b; Tworkowski et al., 2023; Chen et al., 2023; Ratner et al., 2023, *inter alia*) have begun to overcome this limitation by equipping models with the ability to process longer texts, laying the foundation for increasing the number of models involved in communication. In addition, our experiments indicate that model communication can achieve effective performance with reduced computational resources, aligning with the sustainable development goals of AI community (Van Wynsberghe, 2021; Wu et al., 2022).

Furthermore, the concept of AI learning from each other to foster collective improvement is a focal point of current research (Bai et al., 2022b; Ponnusamy et al., 2022; Lee et al., 2023). Our aim and aspiration is to cultivate a collective intelligence among large language models (Ha and Tang, 2022). This approach not only optimizes individual model performance but also contributes to the broader AI research community's pursuit of more advanced, collaborative AI systems.

B Datasets and Evaluation Metrics

Datasets In Table 2, we meticulously detail the specifics and statistics of each dataset employed in

Dataset	Reasoning Task	Ans Type	# Prompt	# Test	License
GSM8K (Cobbe et al., 2021)	Mathematical	Number	8	1,319	MIT License
MultiArith (Roy and Roth, 2015)	Mathematical	Number	8	600	Unspecified
SingleEq (Koncel-Kedziorski et al., 2016)	Mathematical	Number	8	508	Unspecified
AddSub (Hosseini et al., 2014)	Mathematical	Number	8	395	Unspecified
AQUA (Ling et al., 2017)	Mathematical	Multi-choice	4	254	Apache-2.0
SVAMP (Patel et al., 2021)	Mathematical	Number	8	1,000	MIT License
CommonsenseQA (Talmor et al., 2019)	Commonsense	Multi-choice	7	1,221	Unspecified
StrategyQA (Geva et al., 2021)	Commonsense	T/F	6	2,290	MIT license
Date Understanding (Suzgun et al., 2023)	Symbolic	Multi-choice	3	369	MIT license
Penguins in a Table (Suzgun et al., 2023)	Symbolic	Multi-choice	3	146	MIT license

Table 2: Detailed statistics of the datasets utilized in our experiment. Ans Type indicates the form of the answer. # Prompt represent the count of chain-of-thought exemplars employed as few-shot prompts for each task. # Test indicates the quantity of samples contained within each dataset.

our experiments. This includes the data source, task type, answer type, the number of prompt samples used, the total number of test samples, as well as the licenses pertaining to each dataset.

Evaluation Metrics Accuracy is used as the metric for evaluation in our study. For datasets where the answer is numerical, we employ regular expressions to extract the number following the phrase "the answer is" and perform a numerical comparison with the provided answer. For datasets with multiple-choice and true/false questions, accuracy is calculated by checking if the option extracted from the response matches the correct answer.

In the main experiment, all test samples are used for evaluation. In the analysis part, due to rate limits and cost considerations, we set an upper limit on the sample size. Consequently, a maximum of 1,000 samples are utilized for each run.

C Implementation Details

Confidence Evaluation. Considering that confidence evaluation requires historical answers for reference, we begin incorporating the confidence information into the prompts from the second round of communication. Specifically, after calculating C_i using the method described in Section 4.4, we preface the solution with "$\mathcal{M}_i$'s confidence in this solution is $\mathcal{C}_i$", where $\mathcal{M}_i$ is the character name.

Termination Condition. For the consistent output termination condition, a minimum of two rounds of communication is necessary, as it requires the model's answer from the previous round. Given that only three models are involved in the EoT communication, the exit of a single model reduces the interaction to a dialogue between the remaining two, potentially impeding their communication. Therefore, if a single model exits, we terminate the communication and select the exiting model's answer as the final result.

In the case of majority consensus termination, if the answers from all three models align in the first round, we deem further communication unnecessary and end the exchange. Given that only three models are involved in the communication, an exit based on two models holding the same incorrect answer could lead to an inaccurate final result. Therefore, during the initial five rounds, we require a unanimous agreement among all models before ceasing communication. If a consensus is not reached after five rounds, the majority answer will be adopted as the final outcome.

Computation Cost. Computational costs are calculated based on OpenAI's official pricing for `GPT-3.5-Turbo-0301`, which is computed as Input Tokens $\times$ $0.0015/1000$ + Output Tokens $\times$ $0.002/1000$.

D EoT Prompts

During the EoT communication process, we assign different roles to the models. Table 3 displays the prompts for each role, wherein we have models A, B, and C take on the personas of Kitty, Ben, and Peter, three high school students, to facilitate the communication. The specific prompts for different datasets can be found in our Github repository.

E Case Studies

To deepen our understanding of the four communication paradigms, we conducted case studies for each. The processes of these paradigms are detailed in Tables 4, 5, 6, and 7, respectively. These demonstrate that the EoT method, by introducing external insights through cross-model communication, can effectively correct reasoning errors and assist models in arriving at correct answer.

Character Prompts

Kitty: “You are Kitty, a high school student admired for your attentiveness and detail-oriented nature. Your friends often rely on you to catch details they might have missed in their work. Your task is to carefully analyze the presented math problem, apply your attentive skills, and piece together a detailed solution. Afterward, you’ll have the opportunity to review the solutions provided by your friends, offering insights and suggestions. Your careful revisions will help all of you to enhance your understanding and arrive at the most accurate solutions possible.”

Ben: “You are Ben, a high school student with a track record of excellent grades, particularly in mathematics. Your friends admire your diligence and often seek your guidance in their studies. Your role is to scrutinize the problem at hand with your usual attention to detail, drawing from your vast knowledge of math principles. After considering your friends’ approaches, carefully construct your answer, ensuring to clarify each step of your process. Your clear and logical explanations are valuable, as they will serve as a benchmark for your friends to compare and refine their own solutions.”

Peter: “You are Peter, a high school student recognized for your unique problem-solving abilities. Your peers often turn to you for assistance when they encounter challenging tasks, as they appreciate your knack for devising creative solutions. Today, your challenge is to dissect the given math problem, leveraging your unique problem-solving strategies. Once you’ve crafted your solution, share it with your friends, Ben and Kitty, so they can see a different perspective. Your innovative approach will not only provide an answer but also inspire Ben and Kitty to think outside the box and possibly revise their own solutions.”

Communication Prompts

Please consider the example provided and think it step by step.
Question: { }
Here is a solution process from your friend:
Solution: { }
Your friend’s confidence in this solution is: { }
Based on your friend’s solution, carefully re-examine your previous answer. If your friend’s confidence level is below 0.5, it suggests a high probability that the solution might be incorrect. Remember, solutions with high confidence can also be wrong. Utilize your talent and critical thinking to provide a new step-by-step solution process.

Table 3: Examples of system prompts used for role-playing and communication prompts in EoT communication.

Q: Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How load does it take to download the file?

Kitty's Solution: Carla was able to download for 40% of the time before the restart. 40% of 200 GB is 80 GB. Since she can download 2 GB/minute, it would take her 80/2 = 40 minutes to download 80 GB. After the forced restart, she has to restart the download from the beginning, so she would need another 200/2 = 100 minutes to complete the download. Plus the 20 minutes for the restart, the total time would be 40 + 100 + 20 = 160 minutes. So the answer is 160 minutes.

Ben's Solution: Carla needs to download a 200 GB file. Normally, she can download 2 GB/minute, which means it would take her 200 / 2 = 100 minutes to download the file. However, 40% of the way through the download, Windows forces a restart for 20 minutes, which means Carla has downloaded 200 * 0.4 = 80 GB before the restart. After the restart, she needs to start the download again from the beginning, so the total time it takes her would be 100 + (200 / 2) = 200 minutes. Therefore, it takes Carla 200 minutes to download the file. So the answer is 200 minutes.

Peter's Response: Upon re-evaluating the problem and reviewing Kitty and Ben's solutions, it's clear their results deviated due to a misunderstanding of the problem conditions - primarily, how much of the downloaded file was retained after Window's forced restart. To clarify, here is the new step-by-step solution:
1. Start with Carla's download rate: she can download 2 GB/minute.
2. Therefore, in theory, if uninterrupted, Carla would be able to download the 200 GB file in 200 GB / 2 GB/minute = 100 minutes.
3. However, 40% of the way through the download, Windows forced a restart, so Carla had already downloaded 0.40 x 200 GB = 80 GB at that time. This took 80 GB / 2 GB/minute = 40 minutes.
4. The problem statement clarifies that after the restart, Carla has to start again from the beginning. This means the 80 GB downloaded before the forced restart was wasted.
5. Consequently, after the restart, Carla still needs to download the whole 200 GB file, which, at her rate, will take her another 200 GB / 2 GB/minute = 100 minutes.
6. We cannot forget the time for the Windows restart: it took an additional 20 minutes.
7. Consequently, the total time to download the file, taking into account the restart, is the sum of the time for the initial download attempt, the time for the restart, and the time for the full download post-restart. It results in 40 minutes (initial incomplete download) + 20 minutes (Windows restart) + 100 minutes (full download post-restart) = 160 minutes.
From these calculations, it appears that Kitty's solution is correct. It takes Carla 160 minutes to download the file, given the interruption due to Windows update that forced her to restart the download from the beginning. So the answer is 160 minutes.

Table 4: An illustrative example of the Memory paradigm of EoT in the GSM8K dataset.

Q: Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How load does it take to download the file?

Ben's Solution: Carla needs to download a 200 GB file. Normally, she can download 2 GB/minute, which means it would take her 200 / 2 = 100 minutes to download the file. However, 40% of the way through the download, Windows forces a restart for 20 minutes, which means Carla has downloaded 200 * 0.4 = 80 GB before the restart. After the restart, she needs to start the download again from the beginning, so the total time it takes her would be 100 + (200 / 2) = 200 minutes. Therefore, it takes Carla 200 minutes to download the file. So the answer is 200 minutes.

Peter's Solution: Carla can download 2 GB in 1 minute, so to download 200 GB, it will take her 200/2 = 100 minutes if she didn't have to restart. However, she was 40% of the way through the download when Windows forced the restart, so she had already downloaded 0.4 x 200 = 80 GB. The restart took 20 minutes, so she lost 20 x 2 = 40 GB of progress. After the restart, she had to start again from 0 GB. So in total, she had to download 200 - 80 - 40 = 80 GB. At her regular download speed of 2 GB/minute, it will take her 80/2 = 40 minutes to download the remaining 80 GB. Adding the 20 minutes of the Windows update, the total time it takes Carla to download the file is 20 + 40 = 60 minutes. So the answer is 60 minutes.

Kitty's Solution: Carla was able to download for 40% of the time before the restart. 40% of 200 GB is 80 GB. Since she can download 2 GB/minute, it would take her 80/2 = 40 minutes to download 80 GB. After the forced restart, she has to restart the download from the beginning, so she would need another 200/2 = 100 minutes to complete the download. Plus the 20 minutes for the restart, the total time would be 40 + 100 + 20 = 160 minutes. So the answer is 160 minutes.

Kitty's Response: Analyzing both Ben's solution and Peter's solution, Peter's conclusion is incorrect. He misunderstood the situation. After the Windows restart, the download does not continue from where it stopped. It starts from the beginning, which means the download progress does not reduce what is left to download.
Revisiting Ben's solution, he almost got everything right except the final calculation step whereas he added an extra 100 minutes. The correct calculation should be:
Before forced restart: 80 GB (40% of the total file) in 40 minutes (80/2 = 40)
After forced restart: 200 GB total file in 100 minutes (200/2 = 100)
Windows restart: 20 minutes
Total download time = Before restart time + After restart time + Windows restart time
Total download time = 40 minutes + 100 minutes + 20 minutes = 160 minutes
Ben's setup was correct, but he added an extra 100 minutes due to an arithmetic mistake. The download time in his calculations should total to 160 minutes, not 200 minutes. So, effectively upon examining both scenarios by Peter and Ben, when taking all aspects into account and considering the restart, the total time it should take Carla to download the file is 160 minutes in total. So the answer is 160 minutes.

Table 5: An illustrative example of the Report paradigm of EoT in the GSM8K dataset.

Ben's Solution: Carla needs to download a 200 GB file. Normally, she can download 2 GB/minute, which means it would take her 200 / 2 = 100 minutes to download the file. However, 40% of the way through the download, Windows forces a restart for 20 minutes, which means Carla has downloaded 200 * 0.4 = 80 GB before the restart. After the restart, she needs to start the download again from the beginning, so the total time it takes her would be 100 + (200 / 2) = 200 minutes. Therefore, it takes Carla 200 minutes to download the file. So the answer is 200 minutes.

Kitty's Response: After reviewing Ben's solution, it appears there seems to be a mathematical error in the final part of his solution. While he correctly calculates that Carla has downloaded 80 GB of the file before the restart, he fails to account for the time Carla already spent downloading the 80GB in his final total. Furthermore, he miscalculated the time it takes Carla to download the file after the restart. Now, let's correct the calculation. Before the restart, Carla downloaded 40% of the file, which is 40/100 * 200 = 80 GB. Given she downloads 2 GB per minute, this would have taken her 80 / 2 = 40 minutes. Unfortunately, the computer restarts, wasting 20 minutes of her time. After the restart, Carla has to download the entire file again. With her download speed, this will take another 200 / 2 = 100 minutes. Therefore, the total time Carla spends downloading the file is the time before the restart + the time wasted on the restart + the time after the restart, which is 40 minutes + 20 minutes + 100 minutes = 160 minutes. So, the final and correct answer should indeed be 160 minutes.

Peter's Solution: Carla can download 2 GB in 1 minute, so to download 200 GB, it will take her 200/2 = 100 minutes if she didn't have to restart. However, she was 40% of the way through the download when Windows forced the restart, so she had already downloaded 0.4 x 200 = 80 GB. The restart took 20 minutes, so she lost 20 x 2 = 40 GB of progress. After the restart, she had to start again from 0 GB. So in total, she had to download 200 - 80 - 40 = 80 GB. At her regular download speed of 2 GB/minute, it will take her 80/2 = 40 minutes to download the remaining 80 GB. Adding the 20 minutes of the Windows update, the total time it takes Carla to download the file is 20 + 40 = 60 minutes. So the answer is 60 minutes.

Ben's Response: Upon reviewing both solutions, it seems there might have been some misunderstandings. As per the problem, Carla can download at 2 GB/minute, and she needs to download a 200 GB file. By this rate, it would take her 200 GB / 2 GB per minute = 100 minutes without any interruptions. However, 40% of the way through the download, Windows force restart happened which lasted for 20 minutes. By that time, she downloaded 40% of her file which is equal 200 GB * 0.4 = 80 GB. The restart not only consumed 20 minutes but also voided her previous download progress, which means after the updates she had to start downloading all over again. So, the total time taken by Carla to download the 200 GB file would be calculated as: Regular download time (200 GB at 2 GB/minute) + Time spent in updates = (200 / 2) + 20 = 100 + 20 = 120 minutes. So the correct total time it takes Carla to download the file is 120 minutes.

Table 6: An illustrative example of the Relay paradigm of EoT in the GSM8K dataset. Kitty firmly maintains her correct answer, while Ben's response begins to waver.

Q: Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How load does it take to download the file?

Peter's Response: Analyzing both solutions, it appears that there is a slight misunderstanding in Ben's solution. He has missed considering the download progress lost due to the forced restart of Windows. Let's modify and re-explain the solution in a way that takes this into account:
Step 1: Calculate the time for her initial progress. Carla was able to download 40% of the file before the interruption. That is, she downloaded 0.4 * 200GB = 80GB. With her speed of 2GB/minute, she used 80/2 = 40 minutes on this initial download.
Step 2: Consider the time lost due to the forced restart. The Windows forced update lasted for 20 minutes, during which time she made no progress on her download.
Step 3: Calculate the remaining time to finish the download. After the forced restart, Carla had to re-download the whole file. It takes her 200GB / 2GB per minute = 100 minutes at her download speed to download the whole file.
Step 4: Calculate the total time. The total time it takes Carla to download the whole file, considering the interruption, is the sum of the time it took for her initial progress, the time lost due to the forced restart, and the time it took for her to finish the download after the restart. This is 40 minutes + 20 minutes + 100 minutes = 160 minutes.
Therefore, the correct answer is 160 minutes. This improved solution accounts for the hiccup caused by the Windows update which Ben did not factor into his original problem-solving strategy.

Ben's Response: Reviewing the problem again, I can see that there are errors in both my solution and my friend Peter's solution. Here is the corrected step-by-step solution:
1. Carla downloaded 40% of the 200 GB file before the system restart. This equals 40/100 * 200 = 80 GB.
2. At a rate of 2 GB/minute, it must have taken her 80 / 2 = 40 minutes to download that 80 GB.
3. Then, the Windows update interrupted the download for 20 minutes.
4. After the restart, she had to start again from the beginning, so she needs again 200 GB/2 GB per minute = 100 minutes to download the full file.
5. If we add up these times, we get the total time Carla spent on the download: 40 + 20 + 100 = 160 minutes.
So, it would take Carla 160 minutes to download the file, correcting both my previous response and Peter's calculations. This solution process allows for the clear understanding of the time taken to download the file considering both the time before and after the interruption. So the answer is 160 minutes.

Table 7: An illustrative example of the Debate paradigm of EoT in the GSM8K dataset. Through mutual discussion, Ben and Peter identify the flaws in their respective solutions and arrive at the correct answer.