Tuning-Free Accountable Intervention for LLM Deployment - A Metacognitive Approach

Zhen Tan[1], Jie Peng[2], Song Wang[3], Lijie Hu[4], Tianlong Chen[5] Huan Liu[1],
[1]Arizona State University
[2]University of Science and Technology of China
[3]University of Virginia
[4]King Abdullah University of Science and Technology
[5]University of North Carolina at Chapel Hill
ztan36@asu.edu, pengjieb@mail.ustc.edu.cn, sw3wv@virginia.edu, lijie.hu@kaust.edu.sa, tianlong@cs.unc.edu,

Abstract

Large Language Models (LLMs) have brought significant advances across various NLP tasks through few-shot or zero-shot prompting, bypassing the need for parameter tuning. However, the "black-box" nature behind their massive parameter sizes increases the "hallucination" concerns, especially in high-stakes applications (e.g., healthcare), where decision mistakes can lead to severe consequences. In contrast, human decision-making relies on complex cognitive processes, such as the ability to sense and adaptively correct mistakes through conceptual understanding. Drawing inspiration from human cognition, we propose an innovative *metacognitive* approach **CLEAR**, to equip LLMs with capabilities for self-aware error identification and correction. Our framework constructs concept-specific sparse subnetworks that indicate decision processes. This provides a novel interface for model *intervention* after deployment. The benefits include: (*i*) at inference time, our metacognitive LLMs can self-consciously identify potential mispredictions with minimum human involvement, (*ii*) the model can self-correct its errors efficiently without additional tuning, and (*iii*) the correction procedure is not only self-explanatory but also user-friendly, enhancing model interpretability and accessibility. With these metacognitive features, our approach pioneers a new path toward trustworthy LLMs. Appendix is given in the arxiv version.

Code — https://github.com/Zhen-Tan-dmml/CLEAR.git.

1 Introduction

Recent years have witnessed impressive achievements of Large Language Models (LLMs) (Raffel et al. 2020; Zhou et al. 2022; OpenAI 2023). However, LLMs could make mistakes due to issues like "hallucination" (McKenna et al. 2023). Such vulnerabilities pose critical challenges for the trustworthy deployment of LLMs in high-stakes settings where errors can lead to severe consequences. For example, in LLM-assisted medical diagnoses (Monajatipoor et al. 2022), a single wrong diagnosis can inflict significant physical and financial harm to the patient.

Despite the significance of this issue, the current literature lacks an effective approach to LLM *intervention* after deployment to fix errors. With *few-shot* or *zero-shot prompting* (Wei et al. 2022; OpenAI 2023), which recently has shown promising results, users can directly query LLMs and point out their mistakes using usually "hand-crafted" prompts. While straightforward, the post-prompting performance of this method remains uncertain. Moreover, it necessitates human expertise both for error identification and prompt design. Another potential method is to *fine-tune* part of the parameters in LLMs (e.g, the final layers) on incorrectly predicted examples (Hardt and Sun 2023). This method not only requires costly human involvement but also risks model overfitting on those examples and "catastrophic forgetting" of prior knowledge. Some initial work (Li et al. 2023) repeatedly performs *activation-level intervention* on all examples for better performance, thus resulting in increased time complexity during inference. Considering these issues, we identify three main challenges for fixing LLM errors after deployment: ❶ The "black-box" nature of LLMs makes it difficult to locate the source of errors within model parameters, thereby impeding targeted intervention. ❷ Correcting errors usually requires domain experts, which limits scalability and automatic error correction.❸ The complexity and large parameter size of LLMs render targeted intervention a prohibitive task.

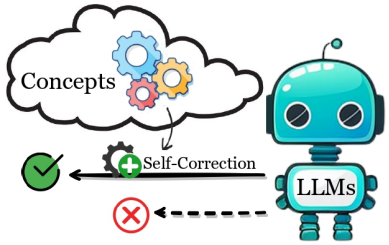


Figure 1: Metacognitive LLMs are able to perceive concepts to self-correct potential errors.

In this paper, we argue that an ideal intervention should be *metacognitive*, where LLMs are capable of self-aware error identification and correction. This perspective is informed by several key insights from cognitive science literature: (*a*) **Cognitive Perception of Concepts** - humans demonstrate the ability to quickly identify and correct judgment errors by recognizing essential features, or "concepts" (Malafouris 2013; Koh et al. 2020). This ability validates the efficiency of human cognitive processes. (*b*) **Neural Sparsity for Efficiency** - building upon the notion of efficiency, the architecture of the human brain is insightful. The distribution of

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

neural connections and activity patterns in brains is highly sparse (Gerum et al. 2020), which facilitates rapid cognitive responses. (*c*) **Conscious Anomaly Detection** - The human brain exhibits an intrinsic ability to consciously identify anomalies or challenging problems (Penfield 2015). Upon encountering such situations, additional neural resources are channeled to address them effectively. Hereby, we provide the definition of metacognition as follows:

Definition 1. ***Metacognition.*** *At inference time, a metacognitive LLM can autonomously detect and correct potential mispredictions in a single run.*

Building on this intuition, we propose the **CLEAR** framework (**C**oncept-**L**earning-**E**nabled met**A**cognitive inte**R**vention) for LLM deployment. CLEAR helps LLMs learn concept-specific sparse subnetworks. These subnetworks elucidate transparent decision-making pathways, thereby providing a unique interface for precise model intervention by automatically allocating more sparse computing modules to potentially more challenging instances. Distinctively, our approach simultaneously tackles the challenges highlighted above via the following contributions:

- ⋆ **Interpretability**. Leveraging the transparency of decision pathways, our **CLEAR** allows for tracing decisions back to the input, thereby aiding user comprehension and trust in the model.
- ⋆ **Efficiency.** When incorrect predictios are identified, the LLM dynamically activates additional internal experts to refine the prediction in a single run without further tuning.
- ⋆ **Effectiveness.** We conduct extensive experiments on real-world datasets with LLM backbones in various sizes and architectures, and the results demonstrate that our intervention consistently improves inference-time predictions.

2 Related Work

Intervention on Deep Models for Error Mitigation. Historically, error mitigation in machine learning emphasized simpler models, such as Decision Trees and Random Forests, where corrections were largely heuristic and human-driven (Doshi-Velez and Kim 2017). With the evolution of machine learning techniques, there was a pivot towards leveraging algorithms themselves for error detection, emphasizing the removal of non-relevant data and unveiling crucial fault-application relationships (Abich et al. 2021). The ascendance of neural networks, and LLMs in particular, brought new intervention paradigms. Fine-tuning emerged as a primary strategy for addressing model shortcomings despite its challenges related to overfitting and catastrophic forgetting of prior knowledge (Wang et al. 2019; French 1999). Few-shot and Zero-shot prompting marked another avenue, guiding models without altering their internal makeup, leading to inherent limitations in error repeatability (Wei et al. 2022; Huang et al. 2023). Deeper interventions targeting model architectures have delivered promising accuracy, yet with computational trade-offs (Li et al. 2023). Notably, quantum error mitigation approaches, though out of our current scope, underline the breadth of exploration in this domain (Subramanian Ravi et al. 2021).

Concurrently, the push towards model interpretability has intensified (Carvalho, Pereira, and Cardoso 2019; Yuksekgonul, Wang, and Zou 2022). The ultimate goal is to design systems whose inner workings can be easily understood, thereby facilitating targeted interventions. Another series of recent work on concept bottleneck models (Koh et al. 2020; Zarlenga et al. 2022) utilize extra human-comprehensive concept labels to guide the learning of LLMs. Those concepts can be annotated by either human (Yuksekgonul, Wang, and Zou 2022; Wu et al. 2022) or large foundation models (Oikarinen et al. 2022; Tan et al. 2023b). However, those methods cannot provide transparency inside the LLM backbone, thus demanding specialized interventions that are usually handcrafted by domain experts (Farrell 2021; Monajatipoor et al. 2022; Tan et al. 2023a).

Metacognitive Approaches. Metacognition, commonly known as "thinking about thinking", has long been recognized in cognitive science (Flavell 1979) through educational and clinical paradigms (Zimmerman 2013; Moritz and Woodward 2007). This foundational knowledge has been applied to AI, aspiring towards machines with self-reflective and adaptive capabilities (Cox 2005). Recent endeavors strive to infuse cognitive inspirations into models, demonstrating a deeper "understanding" of their decisions (Malafouris 2013). However, genuinely metacognitive LLMs remain a difficult goal (Huang et al. 2023), with challenges arising from their black-box nature and vast, intricate architectures.

3 Methodology

The proposed framework **C**oncept-**L**earning-**E**nabled met**A**cognitive inte**R**vention, **CLEAR**, is comprised of two crucial components: (1) *Concept Learning*: the learning of concept-specific sparse subnetworks for LLMs. (2) *Metacognitive Intervention*: automatic error identification and rectification. The core idea of our method is that a refined understanding of LLMs can facilitate targeted metacognitive intervention. To this end, before detailing the proposed CLEAR framework, we first discuss how to learn concept-specific sparse subnetworks.

3.1 Concept Learning for LLMs

Basic Setup. Our primary focus is the enhancement of Large Language Models (LLMs) within the realm of text classification tasks during the inference phase. Given a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{c}^{(i)})_{i=1}^{N}\}$, we utilize an LLM, denoted by $f_{\boldsymbol{\theta}}$, to transform an input text $\boldsymbol{x} \in \mathbb{R}^D$ into a latent space representation $\boldsymbol{z} \in \mathbb{R}^E$. This latent representation is then classified via a linear classifier $g_{\boldsymbol{\phi}}$ into the respective target label y (discrete for classification and continuous for regression). Here $\{\boldsymbol{c}^{(i)}\}_{i=1}^{N}$ denotes the critical features, or "concepts" annotated by humans (Koh et al. 2020; Abraham et al. 2022) or very large language models (Tan et al. 2023b; Ludan et al. 2023), such as GPT-4 (OpenAI 2023). These concepts are represented by one-hot vectors. For instance, in a restaurant review sentiment dataset, the concept "Food" is denoted by $[0, 0, 1]$, signifying a "Positive" attitude towards food. The other vector positions represent "Negative" and "Unknown".

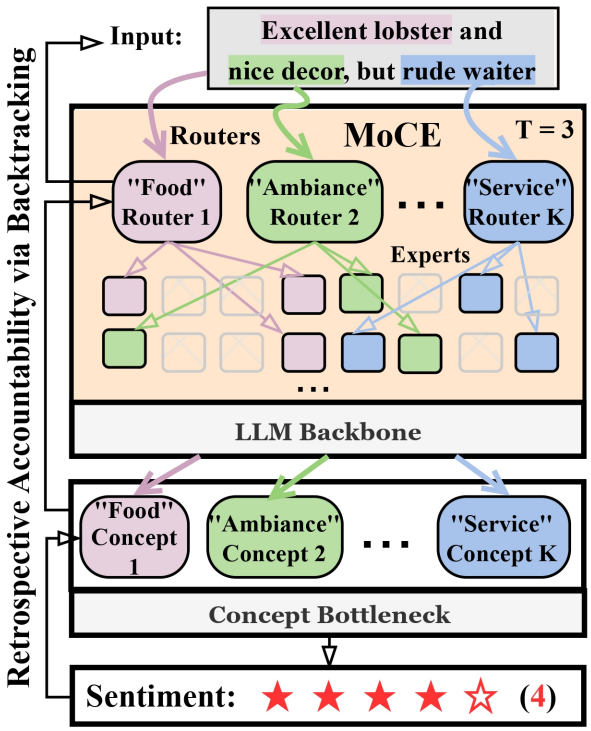


Figure 2: The illustration of the first component *Concept Learninng* of **CLEAR**. During Concept Learning, the LLM backbone learns to construct concept-specific sparse networks via MoCE.

Incorporating Concept Bottlenecks for LLMs. Our general pipeline is inspired by a previous work (Koh et al. 2020) on image classifications. Instead of altering LLM encoders $f_{\boldsymbol{\theta}}$—which might compromise the integrity of the text representation—we incorporate a linear layer, characterized by a sigmoid activation function $p_{\boldsymbol{\psi}}$. This layer maps the latent representation $\boldsymbol{z} \in \mathbb{R}^E$ to a concept space $\boldsymbol{c} \in \mathbb{R}^K$, and then a white-box linear model g_ϕ maps the concepts to the target label y. This creates a decision-making pathway depicted as $\boldsymbol{x} \rightarrow \boldsymbol{z} \rightarrow \boldsymbol{c} \rightarrow y$. By allowing for multi-class concepts, we aim to achieve nuanced interpretations. Akin to common practice (Koh et al. 2020; Tan et al. 2023b), the joint optimization harmonizes the concept encoder and label predictor via weighted sum, represented as $\mathcal{L}_{\text{joint}}$, as detailed in Appendix A.

Building Concept-Specific Sparse Subnetworks via Mixture of Concept Experts. We present the *Mixture of Concept Experts* (MoCE) framework, a novel approach to creating pathways anchored in specific concepts, thereby enhancing targeted interventions, based on the mixture-of-expert (MoE) paradigm (Shazeer et al. 2017) known for the dynamic utilization of unique experts per input. Our motivation here is to leverage the ability of MoE to adaptively select and activate only the most relevant experts (i.e., subnetworks in our work) based on the input's concepts. As such, by conditioning on concept-based computation, MoCE crafts sparse modules, fine-tuning the encoding of text inputs as per their inherent concepts.

We structure blocks of MoCEs as the expert layer. This layer comprises a multi-head attention block combined with multiple parallel experts. Specifically, we adapt MoCE for Transformer architectures, integrating MoE layers within successive Transformer blocks. Crafting a MoCE expert involves segmenting the conventional MLP of transformers into more compact segments (Zhang et al. 2021) or duplicating the MLP (Fedus, Zoph, and Shazeer 2022). Note that the majority of extant MoE studies have predominantly focused on the MLP segment within transformers. This focus arises because MLPs account for approximately two-thirds of the entire model parameters, serving as key repositories of accrued knowledge within memory networks (Geva et al. 2020; Dai et al. 2022). The experts can be symbolized as $\{\boldsymbol{e}_m\}_{m=1}^M$, where m signifies the expert index and M is the total count of experts. For each concept c_k, an **auxiliary routing mechanism**, dubbed $\boldsymbol{r}_k(\cdot)$, is deployed. This mechanism identifies the top-T experts based on peak scores $\boldsymbol{r}_k(x)_m$, with x representing the present intermediate input embedding. Generally, T is much smaller than N, which demonstrate the sparse activations among modules of the LLM backbone, making the inference of the model more efficient. The output, x', emanating from the expert layer is:

$$\begin{aligned} \boldsymbol{x}' &= \sum_{k=1}^{K} \sum_{m=1}^{T} \boldsymbol{r}_k(\boldsymbol{x})_m \cdot \boldsymbol{e}_m(\boldsymbol{x}); \\ \boldsymbol{r}_k(\boldsymbol{x}) &= \texttt{top-T}(\texttt{softmax}(\zeta(\boldsymbol{x})), T), \end{aligned} \tag{1}$$

where ζ is a shallow MLP representing learnable routers (Fedus, Zoph, and Shazeer 2022). For the kth concept, the expert $\boldsymbol{e}_t(\cdot)$ initially processes the given features, after which the router amplifies it using coefficient $\boldsymbol{r}_k(\boldsymbol{x})_t$. The combined embeddings across concepts yield the output $\boldsymbol{x}'$. The $\texttt{top-T}$ operation retains the top T values, nullifying the others. Typically, a balancing mechanism, such as load or importance balancing loss (Shazeer et al. 2017), is implemented to avert the risk of representation collapse, preventing the system from repetitively selecting the same experts across diverse inputs. Transitioning to matrix representation for all MoE layers in the LLM structure, we derive:

$$\begin{aligned} \hat{y} &= \sum_{k=1}^{K} \boldsymbol{\phi}_k \cdot \sigma(\boldsymbol{\psi}_k \cdot f_{\boldsymbol{\theta}_k}(\boldsymbol{x})) \\ &= \sum_{k=1}^{K} \boldsymbol{\phi}_k \cdot \sigma(\boldsymbol{\psi}_k \cdot \sum_{m=1}^{T} \boldsymbol{R}_k(\boldsymbol{x})_m \cdot \boldsymbol{E}_m(\boldsymbol{x})), \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid activation, with $\boldsymbol{R}(\cdot)$ and $\boldsymbol{E}(\cdot)$ symbolizing matrix incarnations of all expert layer routers and experts. Equation (2) portrays a factorized decision trajectory for model prediction. This can be optimized through a single backward iteration of the composite loss as outlined in Equation (1). Equation (2) accomplishes a **core objective**: during inference, the LLM's final prediction intrinsically relies on

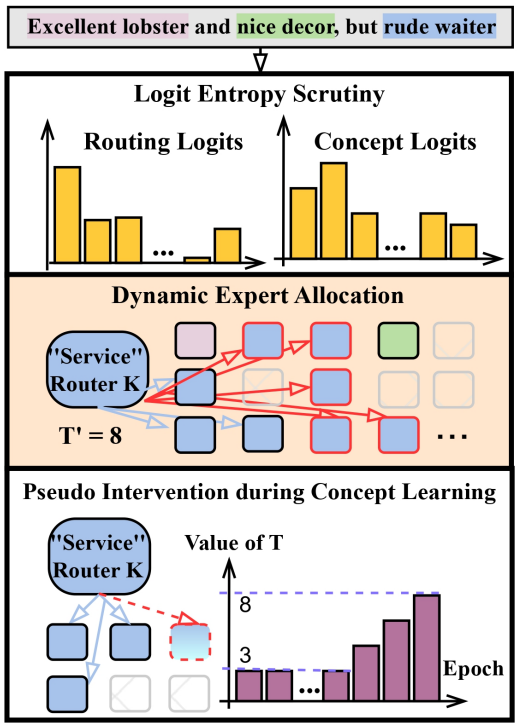


Figure 3: The illustration of the second component *Metacognitive Intervention* of our framework CLEAR, which involves logit entropy scrutiny, dynamic expert allocation, and pseudo intervention, and offers retrospective accountability.

the learned routing policies, the chosen experts, and the perceived concepts. This accountability offers an interface for targeted error identification and interventions.

3.2 Tuning-free Metacognitive Intervention

The *rationale* of our metacognitive intervention is that, different data samples pose varying levels of difficulty for LLMs. Drawing inspiration from human cognitive processes—where the brain identifies and navigates potential challenges—our CLEAR framework proactively detects such issues. It strategically allocates additional sparse neural resources, specifically experts, to effectively address these challenges. This dynamic allocation tailors the response to the complexity of each sample, preventing the model from overfitting on simpler tasks and underfitting on more complex ones. Here, we detail how this is implemented through our defined sparse decision pathways, presenting three research questions, **RQ1-3**, to guide our discussion.

> **RQ1:** *How to achieve "metacognition" for intervention on LLMs?*
> **A1:** *By autonomously monitoring anomalous pattern at critical intermediate layers.*

▷ *Logit Entropy Scrutiny.* The foremost goal is to automatically identify potential errors or more complex cases. As inferred from Equation (2), two critical decision-making phases notably impact the ultimate label prediction: (a) the deduced routing $\{\boldsymbol{R}_k(\boldsymbol{x})\}_{k=1}^{K}$ of the final MoCE layer, and (b) the determined concept activation $\hat{\boldsymbol{a}} = \{\hat{a}_k\}_{k=1}^{K} = \boldsymbol{\psi} \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x})$. Intuitively, an elevated entropy of predictive logits denotes a more dispersed distribution over experts or concept options, signifying lower model confidence and pinpointing instances that deserve additional attention. For this purpose, the Shannon entropy is utilized for logits within the routing and concept activation:

$$H(\boldsymbol{p}) = -\sum_{j=1}^{J} \texttt{softmax}(l_j) \log(\texttt{softmax}(l_j)), \quad (3)$$

where j iterates through the logits' space ($J = M$ for routing and $J = K$ for concept activation). For illustration, the distributions of logits and entropy for concept prediction are depicted using kernel density estimation in Figure 4. It is evident that predictions with errors tend to demonstrate lower confidence and augmented entropy, reinforcing our premise. For automation, as we iterate through the concepts, K-Means clustering is employed to divide confidence levels into two clusters (K=2). The subset with lower confidence is considered to stem from the more challenging instances. K-Means offers the advantage of determining thresholds dynamically, eliminating human involvement. If, for a single concept prediction relating to an instance, the confidence levels of both the routing and concept activation surpass the corresponding thresholds, we tag this concept prediction as potentially erro-

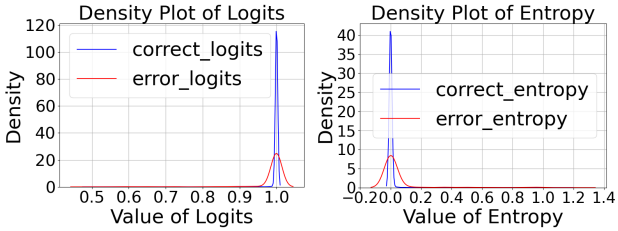
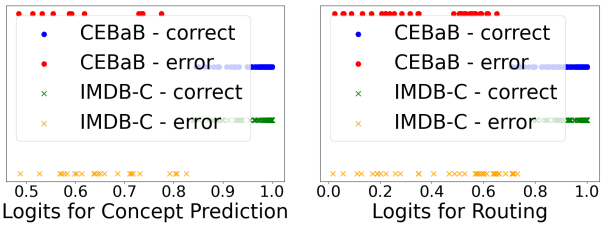


Figure 4: Logit entropy scrutiny. It can be observed that logits of predictions with errors tend to demonstrate lower confidence and larger entropy.



(a) Concept Logits. (b) Routing Logits.

Figure 5: Studies on using K-means for logits scrutiny. This figure illustrates the effectiveness of K-means in distinguishing between correct and erroneous logits for both routing and concept prediction. Logits are normalized via softmax, reducing the impact of noise and extreme values.

neous. We show further studies on the scrutiny for concept and routing are given in Figure 5 (a) and (b). It can be observed that the K-means algorithm effectively distinguishes between correct and incorrect logits.

RQ2: *Once a potential error is identified during inference, how to intervene on LLMs "without extra parameter tuning"?*
A2: *By dynamically allocating experts and enforcing preparatory rehearsal during training.*

▷ *Tuning-free Intervention.* Once an erroneous prediction is identified, we allocate augmented computational resources to secure a more reliable prediction. This operation can be easily achieved by setting the maximum expert number from T to a larger number T' for the router as below. Note that this operation is efficient since no parameter tuning is involved.

$$\boldsymbol{r}_k(\boldsymbol{x}) = \texttt{top-T}(\texttt{softmax}(\zeta(\boldsymbol{x})), T') \tag{4}$$

▷ *Pseudo Intervention during Concept Learning.* Both existing research (Chen et al. 2023) and our experiments (Figure 7 (c) and (d)) indicate that directly adding more experts at the inference stage results in marginal improvements. Drawing inspiration from how humans reinforce understanding of challenging subjects through repeated practice before the final examination, we emulate a similar rehearsal mechanism during concept learning for better metacognitive intervention. As the LLM model is fine-tuned on the task dataset, we progressively raise the count of experts from T to T' linearly after a predetermined number of training epochs, typically after the halfway mark. This strategy of pseudo intervention during the training phase enhances predictions when the expert count is increased during the inference-time intervention. Through this essential rehearsal setup, and by sequentially executing the steps outlined in Equation (3) and Equation (4), the LLM backbone is empowered to autonomously detect possible errors, addressing them more robustly with minimal human involvement.

RQ3: *How can users understand and trust the intervention?*
A3: *By backtracking from the task label, through the sparse pathway, to the input text.*

▷ *Retrospective Accountability.* A standout feature of our metacognitive intervention is its inherent explicability. Using the decision-making pathways showcased in Equation (2), one can trace back from the task label prediction, passing through perceived concepts and activated subnetworks (experts), all the way to the initial text input, as shown in Figure 2. Illustrative examples are provided in Figure 6. The incorporation of our framework, **CLEAR**, represents a harmony of precision, flexibility, and accountability.

4 Experiments

Datasets. Our experiments are conducted on three datasets, including two widely-used real-world datasets, `CEBaB` (Abraham et al. 2022) and `IMDB-C` (Tan et al. 2023b) and a self-curated dataset `ASAP-C`. Each of them is a text *classification* or *regression* dataset comprised of human-annotated concepts and task labels. Their statistics are presented in Table 1. The procedures of curation of the `ASAP-C` dataset are similar to those two existing datasets. More details of datasets are included in Appendix C.

Baselines. For an in-depth analysis, we examine both (*a*) the performance on the *test* sets and (*b*) the performance on the *development* sets, before and after the intervention. This dual-faceted examination allows us to assess the intervention's effectiveness and evaluate the model's potential deterioration in generalizability and catastrophic forgetting of critical prior knowledge. Four LLM backbones are employed in our analysis: BERT (Devlin et al. 2018), OPT (Zhang et al. 2022), and T5 (Raffel et al. 2020). In this study, our evaluation primarily involves two categories of frameworks as baselines. We adjust our choice of LLM backbone per the specific methods employed:
▷ *Direct Intervention Methods*: (*i*) Directly prompting the LLM with human identifying mispredictions. For this method, we use GPT-4 (OpenAI 2023) with zero and few-shot prompting, since it is widely regarded as one of the most capable LLMs currently. (*ii*) Directly fine-tuning the LLM backbones on mispredicted instances identified by humans. (*iii*) Employing the activation intervention method, ITI (Li et al. 2023).
▷ *Concept Bottleneck Models* (CBMs) support concept-level interventions, but still require human experts to identify mispredictions. We consider the following recent CBM frameworks as baselines: (*iv*) Vanilla CBMs (Koh et al. 2020) map the text into concepts using the LLM backbone and involve another linear classifier to perform the final classification. (*v*) Label-free CBMs (LF-CBMs) (Oikarinen et al. 2022) use GPT-4 to obtain the concept labels. (*vi*) Concept embedding models (CEMs) (Zarlenga et al. 2022) that learn continuous embeddings for concepts.

4.1 Superior Performance of CLEAR

Table 2 presents comparative results, averaged over three independent runs, showcasing CLEAR's superiority across concept and task label predictions, for both classification and regression tasks, and at every intervention stage. We adopt an "early stopping" strategy, as per Abraham et al. (2022), to mitigate overfitting, with further details provided in Appendix B and G.

a) **Effectiveness.** CLEAR consistently outperforms baseline models due to its robust MoCE layers, which create sparse, concept-specific subnetworks. This structure not only improves concept internalization but also facilitates effective interventions during inference, resulting in significantly improved prediction accuracy by addressing the challenges specific to each task.

b) **Metacognition**. CLEAR demonstrates critical metacognitive strengths: (*a*) *Efficiency:* Without the need for fine-tuning tuning, CLEAR avoids the common pitfalls of catastrophic forgetting (shaded in gray). (*b*) *Autonomy:* It operates independently of human intervention, which is crucial in scenarios where expertise is scarce. Unlike LF-CBMs that suffer from using noisy labels from GPT-4 (shaded in pink), CLEAR's autonomy emphasizes

Dataset	CEBaB (5-way classification)				IMDB-C (2-way classification)				ASAP-C (regression)			
	Train / Dev / Test		1755 / 1673 / 1685		Train / Dev / Test		100 / 50 / 50		Train / Dev / Test		1005 / 281 / 283	
	Label	Negative	Positive	Unknown	Label	Negative	Positive	Unknown	Label	Negative	Positive	Neutral
Concept	Food	1693 (33.1%)	2087 (40.8%)	1333 (26.1%)	Acting	76 (38%)	66 (33%)	58 (29%)	Content	421 (26.8%)	684 (43.6%)	464 (29.6%)
	Ambiance	787 (15.4%)	994 (19.4%)	3332 (65.2%)	Storyline	80 (40%)	77 (38.5%)	43 (21.5%)	Reasoning	764 (48.7%)	467 (29.8%)	338 (21.5%)
	Service	1249 (24.4%)	1397 (27.3%)	2467 (48.2%)	Emotional Arousal	74 (37%)	73 (36.5%)	53 (26.5%)	Language	382 (24.3%)	569 (36.3%)	618 (39.4%)
	Noise	645 (12.6%)	442 (8.6%)	4026 (78.7%)	Cinematography	118 (59%)	43 (21.5%)	39 (19.4%)	Supportiveness	541 (34.5%)	685 (43.7%)	343 (21.9%)

Table 1: Statistics of experimented datasets and concepts.

Methods	Backbones	CEBaB								IMDB-C							
		Pre-intervention				Post-intervention				Pre-intervention				Post-intervention			
		Dev		Test		Dev		Test		Dev		Test		Dev		Test	
		Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task
Direct Intervention Methods																	
Prompting	GPT4	-	46.52	-	45.87	-	46.52	-	48.32	-	69.35	-	68.74	-	69.35	-	69.84
Fine-tuning	BERT	-	80.03	-	79.75	-	76.43	-	81.23	-	74.52	-	72.11	-	71.69	-	74.26
	OPT	-	82.65	-	81.37	-	80.84	-	82.16	-	80.62	-	79.98	-	75.42	-	81.05
	T5	-	82.64	-	82.65	-	80.67	-	83.34	-	81.85	-	79.87	-	77.62	-	81.53
ITI	T5	-	82.64	-	82.65	-	82.64		83.29	-	81.85	-	79.87	-	81.85	-	81.25
Concept Bottleneck Models																	
Vanilla-CBMs	BERT	85.86	78.32	85.29	78.11	85.86	78.32	88.52	79.52	64.52	72.51	62.76	70.41	64.52	72.51	65.31	71.96
	OPT	87.84	80.03	87.27	79.73	87.84	80.03	89.62	80.12	67.15	78.96	66.53	78.21	67.15	78.96	69.47	79.34
	T5	88.20	81.05	87.96	80.63	88.20	81.05	90.21	81.05	68.85	79.58	67.94	78.26	68.85	79.58	70.26	79.95
LF-CBMs	BERT	82.37	75.24	83.45	75.69	82.37	75.24	83.52	75.82	62.51	70.49	60.35	68.21	62.51	70.49	61.32	68.13
	OPT	84.54	77.62	84.62	76.84	84.54	77.62	85.36	76.64	64.18	75.24	63.37	75.06	64.18	75.24	63.58	74.65
	T5	85.68	78.25	85.74	77.22	85.68	78.25	85.59	76.87	65.16	76.83	64.92	76.30	65.16	76.83	64.43	75.68
CEMs	BERT	86.78	79.10	86.62	78.64	86.78	79.10	88.67	80.04	64.86	72.61	62.84	71.05	64.86	72.61	65.57	72.33
	OPT	87.98	80.51	87.92	79.86	87.98	80.51	89.89	80.65	68.29	79.67	66.97	78.68	67.84	79.62	70.34	79.75
	T5	88.64	81.32	88.34	80.69	88.64	81.32	90.65	81.42	68.98	79.83	68.65	79.64	68.98	79.83	70.93	80.72
Metacognition Intervention																	
CLEAR	OPT-MoCE	88.24	80.96	88.24	80.39	89.04	80.85	90.46	81.24	68.83	79.75	68.47	79.52	68.39	79.86	71.02	80.12
CLEAR	T5-MoCE	89.65	81.62	89.63	81.30	89.65	81.62	91.25	82.14	69.46	80.25	69.65	80.63	69.46	80.25	71.67	80.95

Table 2: Comparative results on the CEBaB and IMDB-C datasets, using *Macro F1* (↑) as the evaluation metric, expressed in percentages (%). Scores shaded in gray highlight instances where the model experienced catastrophic forgetting, leading to a decline in performance on the development set. Scores shaded in pink indicate a decrease in performance following the intervention. Scores shaded in blue are from CLEAR. Results on the ASAP-C dataset in given in Appendix E.

the importance of precise intervention. (*c*) *Accountability:* Through transparent decision-making processes at concept, subnetwork, and input levels, CLEAR thus significantly boosts user trust.

c) **Flexibility.** CLEAR's architecture-agnostic design facilitates its integration with a variety of LLMs, such as OPT and T5, demonstrating its broad adaptability. However, we have not conducted experiments with exceedingly large MoEs like LLaMA-MoE (Team 2023) and Mixtral (Jiang et al. 2024) due to their substantial size, which makes them impractical for training on the datasets. Efficient fine-tuning strategies for these larger models present a promising avenue for future research.

4.2 Extra Investigation and Ablation Study

Accountability. CLEAR excels by offering retrospective interpretability and deep insights into its intervention processes, enhancing transparency at multiple levels. Through backtracking, it provides explanations from the concept, subnetwork, to input levels, significantly increasing user trust and comprehension of the model's decisions.

▷ ***Case Study.*** A case study showcased in Figure 6 (with additional examples in Appendix H) illustrates CLEAR's intervention process. It highlights how CLEAR corrects the predicted label for "Cinematography" from incorrect "-" to correct "+", refining the overall task label. This example, particularly the analysis of activations before and after intervention, uncovers the neural strategy behind CLEAR's corrections, enhancing real-world applicability. For instance, we can compute the influence I of each concept c_k to the final decision by the product of the concept activation $\hat{a}_k$ and the corresponding weight w_k in the linear classifier: $I(c_k) = \hat{a}_k \cdot w_k$, as visualized in Figure 6 (c), demonstrating CLEAR's ability to not only rectify but also explain prediction errors.

Method	Human labels	Parameter tuning	Targeted intervention
Prompting	✔	✘	✘
Fine-tuning	✔	✔	✘
ITI	✘	✘	✘
CBM	✔	✘	✘
CLEAR	✘	✘	✔

Table 3: Efficiency comparison between interventions

Autonomy and Efficiency. CLEAR's autonomy and tuning-free approach distinguish it from other models. As shown in Table 3, CLEAR uniquely improves without requiring human input or complex tuning, a necessity in other models. This independence not only simplifies CLEAR's operation but also heightens its reliability and efficacy, ensuring robustness and trustworthiness.

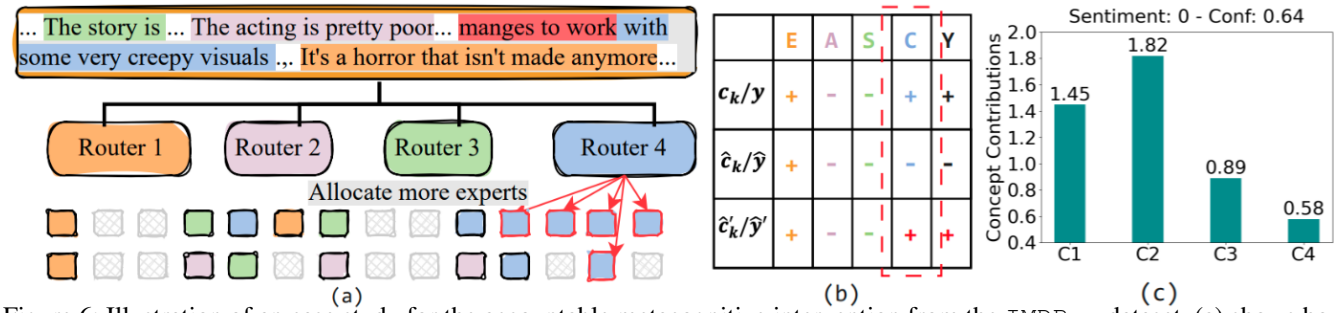
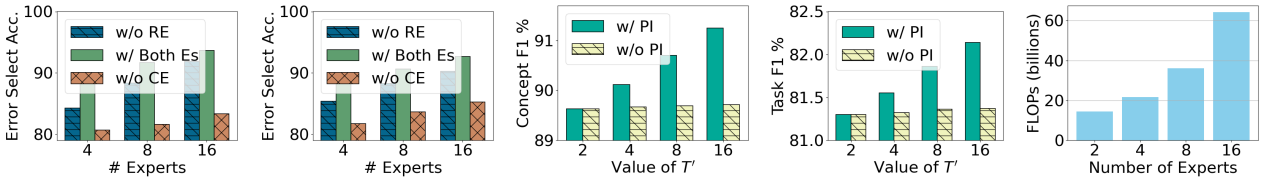


Figure 6: Illustration of an case study for the accountable metacognitive intervention from the IMDB-C dataset. (a) shows how CLEAR performs the intervention by allocating more experts. (b) demonstrates the rectification of the concept label prediction. (c) visualizes the contributions of different concepts.

Methods	CEBaB						IMDB-C						ASAP-C					
	Pre-intervention		Post-intervention		Improvement (↑)		Pre-intervention		Post-intervention		Improvement (↑)		Pre-intervention		Post-intervention		Improvement (↑)	
	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task	Concept	Task
CLEAR (null)	89.63	81.30	89.63	81.30	0	0	69.65	80.63	69.65	80.63	0	0	87.35	0.694	87.35	0.694	0	0
CLEAR (max)	89.63	81.30	86.62	78.81	-3.01	-2.49	69.65	80.63	65.74	78.55	-3.91	-2.08	87.35	0.694	85.34	0.726	-2.01	-0.032
CLEAR	89.63	81.30	91.25	81.80	1.62	0.5	69.65	80.63	71.67	80.95	2.02	0.32	87.35	0.694	89.65	0.624	2.30	0.070
CLEAR (oracle)	89.63	81.30	91.98	82.06	2.35	0.76	69.65	80.63	72.64	81.36	2.99	0.73	87.35	0.694	90.82	0.597	3.47	0.097

Table 4: Ablation study on intervention. "Null" means no intervention is taken. "Max" means directly activate all the experts for all samples. Scores are reported in % and those shaded in pink and blue respectively indicate negative and positive improvements.



(a) CEBaB (b) IMDB-C (c) CEBaB (d) IMDB-C (e) FLOPs analysis

Figure 7: Extra studies on CLEAR. (a) and (b) investigate logit entropies for scrutiny under different expert numbers, where RE denotes *routing entropy*, and CE denotes *concept prediction entropy*. (c) and (d) examine the effects of w/wo *pseudo intervention* (PI) on gradually increased intervention expert number T'. (e) indicates the FLOPs counts *v.s.* expert number. As expected, the results indicate an approximately linear increase in computational complexity with the number of experts.

Ablation Study. We conducted ablation studies to assess CLEAR's core components with each finding detailed below:

- ▷ *Intervention Mechanism.* Table 4 reveals that indiscriminate expert activation for all instances diminishes performance due to overfitting. Comparatively, CLEAR's metacognitive intervention closely matches the precision of oracle interventions using human-annotated labels, validating its effective error correction and metacognitive capacity without human-annotated labels.
- ▷ *Options for Logit Entropy Scrutiny.* Analysis in Figure 7 (a) and (b) shows superior model performance when utilizing both entropy thresholds together rather than separately. Particularly, omitting concept prediction entropy significantly reduces performance, validating CLEAR's design of concept-specific subnetworks that are crucial for its precision in intervention.
- ▷ *Pseudo Intervention.* As demonstrated in Figure 7 (c) and (d), incorporating pseudo intervention markedly improves CLEAR's performance, affirming the strategy of increasing expert numbers during training as a rehearsal enhances preparedness for real-time interventions.
- ▷ *Sensitivity Analysis on the Number of Experts.* Figures 7 (a) and (b) indicate performance boosts with additional experts, attributing to expanded model capacity and learning ability. Furthermore, Figures 7 (c) and (d) demonstrate enhanced accuracy in correcting mispredictions with more experts during the intervention phase.

5 Conclusion

This paper introduces a novel framework, CLEAR, with robust capabilities to autonomously identify and correct errors, thereby reducing the need for extensive human involvement and complicated adjustments. By employing a metacognitive strategy inspired by human cognitive processes, CLEAR enables the construction of transparent, concept-specific sparse subnetworks. This attribute enables clear, comprehensible decision pathways and facilitates post-deployment model intervention. Confronted with the "black-box" issue prevalent in LLMs, CLEAR demonstrates its effectiveness in reducing mispredictions and enhancing overall model interpretability and accessibility. These advances by CLEAR manifest an enhancement in both the performance and reliability of LLMs, ensuring their more trustworthy and accountable deployment in diverse real-world scenarios. We hope the application of CLEAR provides a positive shift for trustworthy LLMs.

References

Abich, G.; Garibotti, R.; Bandeira, V.; da Rosa, F.; Gava, J.; Bortolon, F.; Medeiros, G.; Moraes, F. G.; Reis, R.; and Ost, L. 2021. Evaluation of the soft error assessment consistency of a JIT-based virtual platform simulator. *IET Computers & Digital Techniques*, 15(2): 125–142.

Abraham, E. D.; D'Oosterlinck, K.; Feder, A.; Gat, Y.; Geiger, A.; Potts, C.; Reichart, R.; and Wu, Z. 2022. CEBaB: Estimating the causal effects of real-world concepts on NLP model behavior. *Advances in Neural Information Processing Systems*, 35: 17582–17596.

Carvalho, D. V.; Pereira, E. M.; and Cardoso, J. S. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8): 832.

Chen, T.; Zhang, Z.; Jaiswal, A.; Liu, S.; and Wang, Z. 2023. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. *arXiv preprint arXiv:2303.01610*.

Cox, M. T. 2005. Metacognition in computation: A selected research review. *Artificial intelligence*, 169(2): 104–141.

Dai, Y.; Tang, D.; Liu, L.; Tan, M.; Zhou, C.; Wang, J.; Feng, Z.; Zhang, F.; Hu, X.; and Shi, S. 2022. One model, multiple modalities: A sparsely activated approach for text, sound, image, video and code. *arXiv preprint arXiv:2205.06126*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Farrell, C.-J. 2021. Identifying mislabelled samples: machine learning models exceed human performance. *Annals of Clinical Biochemistry*, 58(6): 650–652.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1): 5232–5270.

Flavell, J. H. 1979. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10): 906.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4): 128–135.

Gerum, R. C.; Erpenbeck, A.; Krauss, P.; and Schilling, A. 2020. Sparsity through evolutionary pruning prevents neuronal networks from overfitting. *Neural Networks*, 128: 305–312.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Hardt, M.; and Sun, Y. 2023. Test-Time Training on Nearest Neighbors for Large Language Models. *arXiv preprint arXiv:2305.18466*.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept bottleneck models. In *International Conference on Machine Learning*, 5338–5348. PMLR.

Li, K.; Patel, O.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *arXiv preprint arXiv:2306.03341*.

Ludan, J. M.; Lyu, Q.; Yang, Y.; Dugan, L.; Yatskar, M.; and Callison-Burch, C. 2023. Interpretable-by-Design Text Classification with Iteratively Generated Concept Bottleneck. *arXiv preprint arXiv:2310.19660*.

Malafouris, L. 2013. *How things shape the mind*. MIT press.

McKenna, N.; Li, T.; Cheng, L.; Hosseini, M. J.; Johnson, M.; and Steedman, M. 2023. Sources of Hallucination by Large Language Models on Inference Tasks. *arXiv preprint arXiv:2305.14552*.

Monajatipoor, M.; Rouhsedaghat, M.; Li, L. H.; Jay Kuo, C.-C.; Chien, A.; and Chang, K.-W. 2022. Berthop: An effective vision-and-language model for chest x-ray disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 725–734. Springer.

Moritz, S.; and Woodward, T. S. 2007. Metacognitive training for schizophrenia patients (MCT): a pilot study on feasibility, treatment adherence, and subjective efficacy. *German Journal of Psychiatry*, 10(3): 69–78.

Oikarinen, T.; Das, S.; Nguyen, L. M.; and Weng, T.-W. 2022. Label-free Concept Bottleneck Models. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Penfield, W. 2015. *Mystery of the mind: A critical study of consciousness and the human brain*. Princeton University Press.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Subramanian Ravi, G.; Smith, K. N.; Gokhale, P.; Mari, A.; Earnest, N.; Javadi-Abhari, A.; and Chong, F. T. 2021. VAQEM: A Variational Approach to Quantum Error Mitigation. *arXiv e-prints*, arXiv–2112.

Tan, Z.; Chen, T.; Zhang, Z.; and Liu, H. 2023a. Sparsity-guided holistic explanation for llms with interpretable inference-time intervention. *arXiv preprint arXiv:2312.15033*.

Tan, Z.; Cheng, L.; Wang, S.; Bo, Y.; Li, J.; and Liu, H. 2023b. Interpreting Pretrained Language Models via Concept Bottlenecks. *arXiv preprint arXiv:2311.05014*.

Team, L.-M. 2023. LLaMA-MoE: Building Mixture-of-Experts from LLaMA with Continual Pre-training.

Wang, H.; Focke, C.; Sylvester, R.; Mishra, N.; and Wang, W. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Wu, Z.; D'Oosterlinck, K.; Geiger, A.; Zur, A.; and Potts, C. 2022. Causal Proxy Models for Concept-Based Model Explanations. *arXiv preprint arXiv:2209.14279*.

Yuksekgonul, M.; Wang, M.; and Zou, J. 2022. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*.

Zarlenga, M. E.; Barbiero, P.; Ciravegna, G.; Marra, G.; Giannini, F.; Diligenti, M.; Precioso, F.; Melacci, S.; Weller, A.; Lio, P.; et al. 2022. Concept Embedding Models. In *NeurIPS 2022-36th Conference on Neural Information Processing Systems*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, Z.; Lin, Y.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2021. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*, 13.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2022. Large Language Models are Human-Level Prompt Engineers. In *The Eleventh International Conference on Learning Representations*.

Zimmerman, B. J. 2013. Theories of self-regulated learning and academic achievement: An overview and analysis. *Self-regulated learning and academic achievement*, 1–36.