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

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have catalyzed transformative advances across a spectrum of natural language processing tasks through few-shot or zero-shot prompting, bypassing the need for parameter tuning. While convenient, this modus operandi aggravates "hallucination" concerns, particularly given the enigmatic "black-box" nature behind their gigantic model sizes. Such concerns are exacerbated in high-stakes applications (e.g., healthcare), where unaccountable decision errors can lead to devastating consequences. In contrast, human decision-making relies on nuanced cognitive processes, such as the ability to sense and adaptively correct misjudgments through conceptual understanding. Drawing inspiration from human cognition, we propose an innovative *metacognitive* approach, dubbed **CLEAR**, to equip LLMs with capabilities for self-aware error identification and correction. Our framework facilitates the construction of concept-specific sparse subnetworks that illuminate transparent decision pathways. This provides a novel interface for model *intervention* after deployment. Our intervention offers compelling advantages: (*i*) at deployment or inference time, our metacognitive LLMs can self-consciously identify potential mispredictions with minimum human involvement, (*ii*) the model has the capability to self-correct its errors efficiently, obviating the need for additional tuning, and (*iii*) the rectification procedure is not only self-explanatory but also user-friendly, enhancing the interpretability and accessibility of the model. By integrating these metacognitive features, our approach pioneers a new path toward engendering trustworthiness and accountability in the deployment of LLMs.

1 Introduction

Recent years have witnessed laudable achievements of Large Language Models (LLMs) (Raffel et al., 2020; Zhou et al., 2022b; OpenAI, 2023). However, LLMs are not infallible; they err due to factors like "hallucination" (McKenna et al., 2023). These vulnerabilities pose critical challenges for the trustworthy deployment of LLMs in high-stakes settings where errors can precipitate significant repercussions. As an example, for LLM-assisted medical diagnoses (Monajatipoor et al., 2022), a single misdiagnosis can inflict profound physical and financial costs on the patient.

Despite its significance, the current literature lacks an effective approach to LLM *intervention* after deployment to help the model overcome those errors. One intuitive method, *few-shot* or *zero-shot prompting* (Wei et al., 2022; OpenAI, 2023) recently has shown promising results. Users can directly query LLMs and point out their mistakes using usually "hand-crafted" prompts. Though they are simple, the post-prompting performance remains uncertain. Moreover, it necessitates human expertise both for error identification and prompt design. (2) Another potential method is to *fine-tune* part of the parameters in LLMs (*e.g*, the final layers) on erroneously predicted examples (Hardt and Sun, 2023). Besides costly human involvement, this method risks model overfitting on those examples and "catastrophic forgetting" of prior knowledge. (3) Some initial work (Li et al., 2023) repetitively performs *activation-level intervention* on all examples to get better performance, thus resulting in inflated inference latency. Against this backdrop, we trifurcate the challenges for LLM intervention into three folds. ❶ The "black-box" nature of LLMs obscures the malfunction source within the multitude of parameters, impeding targeted intervention. ❷ Rectification typically relies on domain experts to identify errors, hindering scalability and automatic intervention. ❸ The architectural complexity and sheer size of LLMs render targeted intervention a daunting task.

In this paper, we advocate that an ideal intervention should be *metacognitive*, where LLMs

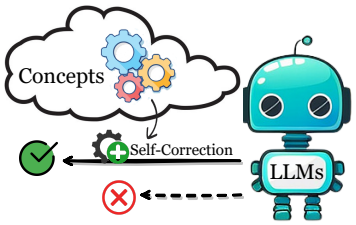


Figure 1: Metacognitive LLMs are able to preceive concepts to self-correct potential errors.

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 swiftly identify and rectify judgment errors by perceptively recognizing essential features, or "concepts" (Malafouris, 2013; Koh et al., 2020). This ability to hone in on vital features underscores the efficiency of human cognitive processes. (*b*) **Neural Sparsity for Efficiency** - building upon the notion of efficiency, the architecture of the human brain provides a valuable lesson. The distribution of neural connections and activity patterns in our brains is characterized by a high degree of sparsity (Gerum et al., 2020). This sparse configuration is believed to facilitate rapid cognitive responses. (*c*) **Conscious Anomaly Detection** - human brain exhibits an intrinsic ability to consciously identify anomalies or challenging problems (Penfield, 2015). Upon encountering such situations, it channels additional neural resources to address them effectively. Building on this premise, we propose an avant-garde **C**oncept-**L**earning-**E**nabled met**A**cognitive inte**R**vention framework, herein termed **CLEAR**, for LLM deployment. CLEAR facilitates LLMs in mastering concept-specific sparse subnetworks. These subnetworks elucidate transparent decision-making pathways, thereby providing a unique interface for surgical model intervention, that automatically allocates more sparse computing modules to potentially more challenging instances. Distinctively, our approach simultaneously tackles the challenges highlighted above through the following four core contributions:

- ⋆ **Metacognition.** At deployment (inference) time, our metacognitive framework autonomously detects potential mispredictions by measuring logit entropy in pivotal intermediate layers.
- ⋆ **Interpretability**. Leveraging the transparency of decision pathways, our **CLEAR** allows for a logical backtrack to the input, thereby aiding user comprehension and fostering trust in the model.
- ⋆ **Efficiency.** Upon identification of a misprediction, the LLM architecture dynamically activates extra internal experts to refine concept perception without necessitating further parameter tuning.
- ⋆ **Effectiveness.** Experiments on real-world datasets with LLM backbones in various sizes and architectures manifest 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 et al., 2019; Yuksekgonul et al., 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 et al., 2022; Wu et al., 2022) or large foundation models (Oikarinen et al., 2022; Tan et al., 2023b). However, those method cannot provide

transparency inside the LLM backbone, thus demanding specialized interventions that are usually hand-carfted 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), resonating through educational and clinical paradigms (Zimmerman, 2013; Moritz and Woodward, 2007). This foundational knowledge has segued into AI, aspiring towards machines with self-reflective and adaptive capabilities (Cox, 2005). Recent endeavors strive to infuse cognitive inspirations into models, affirming a deeper "understanding" of their decisions (Malafouris, 2013). However, genuinely metacognitive LLMs remain an elusive 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. At the heart of our methodology lies the insight that a refined understanding of LLMs can facilitate targeted metacognitive intervention. To this end, before jumping into the full picture of the proposed CLEAR framework, we first explore the learning of 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 typically represented using 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 can represent "Negative" and "Unknown".

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 $g_{\boldsymbol{\phi}}$ maps the concepts to the target label y. This creates a decision-making pathway depicted as $\boldsymbol{x} \to \boldsymbol{z} \to \boldsymbol{c} \to y$. 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 Equation (5) in Appendix A.

Building Concept-Specific Sparse Subnetworks via Mixture of Concept Experts. We presents the *Mixture of Concept Experts* (MoCE) framework, a novel approach to creating pathways anchored in specific concepts, thereby enhancing targeted interventions. This model takes cues from mixture-of-expert (MoE) paradigms (Shazeer et al., 2017), known for their dynamic activation of unique network subsets per input. 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 et al., 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 parameter set, 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.

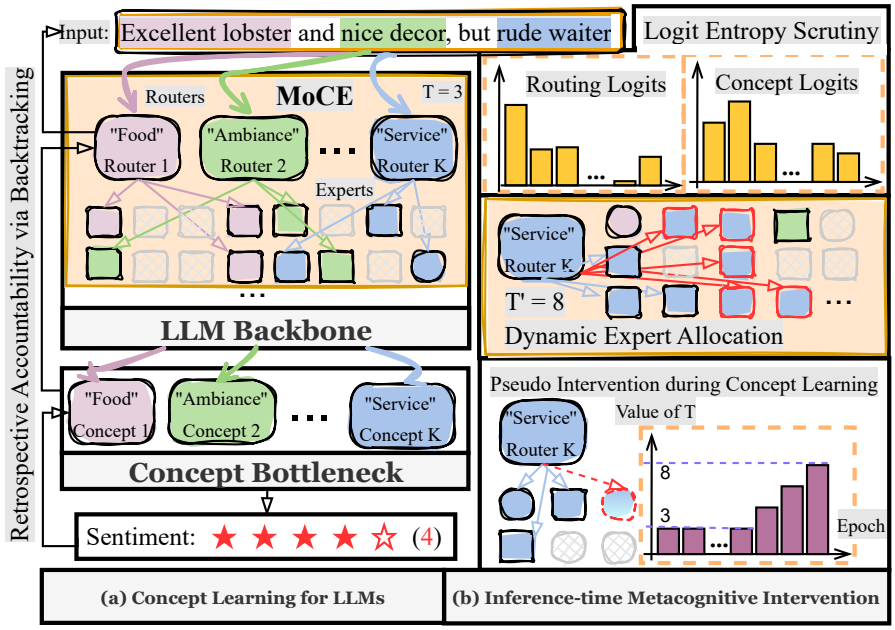


Figure 2: The illustration of the proposed framework **CLEAR**, comprised of two components: (a) *Concept Learinng*, where the LLM backbone learns to construct concept-specific sparse networks via MoCE; and (b) *Metacognitive Intervention*, which involves logit entropy scrutiny, dynamic expert allocation, and pseudo intervention, and offers retrospective accountability.

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 underscores 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 et al., 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 `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 rely on 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.*

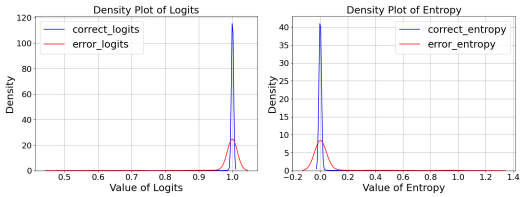
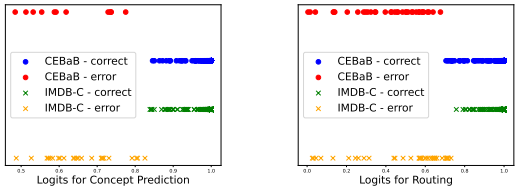


Figure 3: Logit entropy scrutiny. It can be observed that logits of predictions with errors tend to demonstrate lower confidence and larger entropy.



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

Figure 4: 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.

▷ *Logit Entropy Scrutiny.* The foremost goal is to automatically identify potential errors or more complex cases. As inferred from Equation 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:

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

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 3. 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 erroneous. We show further studies on the scrutiny in Figure 4 (a) and (b).

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 very efficient since no extra parameter tuning is involved.

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

▷ *Pseudo Intervention during Concept Learning.* Both existing research (Chen et al., 2023) and our experiments (Figure 6 (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 post the halfway mark. This strategy of pseudo intervention during the training phase significantly enhances predictions when the expert count is increased during the inference-time metacognitive intervention, as depicted in Figure 6 (c) and (d). 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 oversight.

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

Table 1: Statistics of experimented datasets and concepts.

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%)

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

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 Table 6 in Appendix E.

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

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 presents a promising avenue for future research.

4.2 Extra Investigation and Ablation Study

Accoutability. 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 5 (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 5 (c), demonstrating CLEAR's ability to not only rectify but also explain prediction errors.

Table 3: Efficiency comparison between interventions

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

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.

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, underscor-

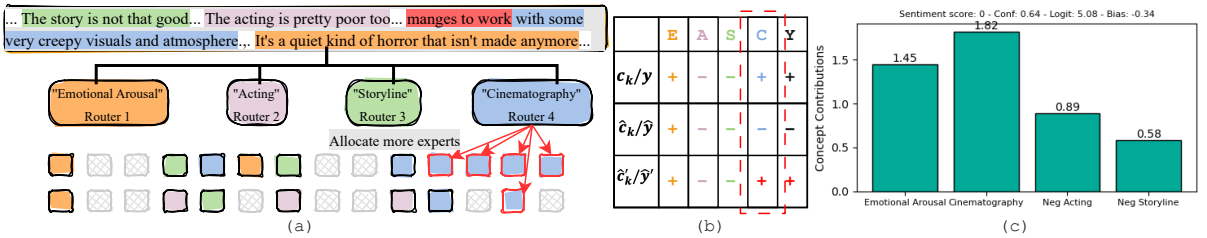
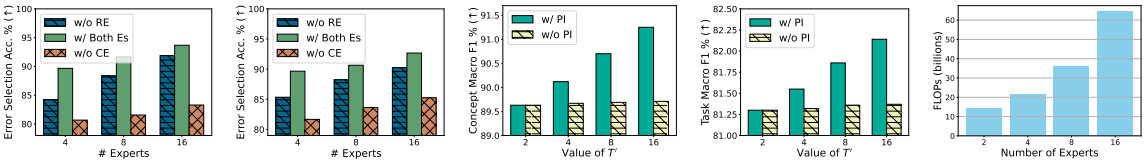


Figure 5: Illustration of an case study for the accountable metacognitive intervention from the IMDB-C dataset. (a) shows how CLEAR perform the intervention by allocating more experts. (b) demonstrates the rectification of the concept label prediction. (c) visualizes the contributions of different concepts.

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

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



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

Figure 6: 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.

ing its effective error correction and metacognitive capacity without human-annotated labels.

▷ *Options for Logit Entropy Scrutiny.* Analysis in Figure 6 (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.* Demonstrated in Figure 6 (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 6 (a) and (b) indicate performance boosts with additional experts, attributing to expanded model capacity and learning ability. Furthermore, Figures 6 (c) and (d) highlight enhanced accuracy in correcting mispredictions with more experts during the intervention phase, underscoring the importance of a higher number of experts throughout CLEAR's operation.

5 Conclusion

This paper outlines a novel framework, CLEAR, in its robust capabilities in autonomously identifying and correcting errors, thereby reducing the need for extensive human oversight and intricate adjustments. By employing a metacognitive strategy inspired by human cognitive processes, CLEAR enables the construction of transparent, concept-specific sparse subnetworks. This attribute ensures clear, comprehensible decision pathways and eases post-deployment model intervention. In tackling the enduring "black-box" issue prevalent in LLMs, CLEAR confidently showcases its effectiveness in diminishing mispredictions and bolstering overall model interpretability and accessibility. These advances by CLEAR underscore a significant enhancement in both the performance and reliability of LLMs, ensuring their more trustworthy and accountable deployment in diverse real-world scenarios. Moving forward, the widespread application of CLEAR promises a tangible, positive shift for safe deployment of LLMs.

Limitations

While our proposed framework, CLEAR, introduces significant advancements in leveraging metacognitive approaches for Large Language Models (LLMs), it is important to acknowledge certain limitations for future research:

1. **Dependency on Pre-defined Concepts:** Inherent from the literature of concept bottleneck models, CLEAR's efficacy also relies on the availability of pre-defined, human-annotated concept labels. This requirement could restrict its application in domains where such labels are scarce or in settings that demand the discovery of emergent concepts. Some existing solution including use other very large langugae model, such as, GPT-4, to get those annotaions (Tan et al., 2023b; Ludan et al., 2023), or using active learning (Tan et al., 2024) or self-training (Singh et al., 2023) to select only subset of samples for concept annotation.

2. **Scalability with Larger Models:** Our current implementation and evaluations primarily focus on models of specific sizes. The adaptability and performance of CLEAR with much larger, particularly proprietary models like GPT-4 or emerging architectures, remain less explored. The reasons include: (1) many of the very large MoE models are proprietary and not accessible for us to train; (2) those models hosting billions of parameters require very large datasets for training, which is not available in the concept learning literature. Even though we have experimented on all available datasets and curated a new dataset for regression task, the amount of tokens remains insufficient for training huge MoE models such as Mixtral (Jiang et al., 2024) and Grok (Corp, 2024) This limitation is indeed a broader challenge within the research community, one that has not been adequately resolved.

3. **Fairness and Bias:** The reliance of CLEAR on pre-defined concepts may inadvertently perpetuate biases present in the training data or annotations. Ensuring fairness and mitigating bias in the model's predictions require careful scrutiny of the data and possibly the incorporation of fairness-aware algorithms. This challenge echoes with the previous mentioned data insufficiency issue. We hope our work can attract more attention in this field and advocate for curating larger-scale opensource datasets.

4. **Adaptation to Dynamic Environments:** The ability of CLEAR to adapt to changing data distributions over time has not been thoroughly examined. Continuous learning environments, where concepts and relationships may evolve, present a critical test for the framework's long-term viability. This can be a potential direction for future work.

Ethical Statement

In developing CLEAR, we have conscientiously considered the ethical implications of our work with Large Language Models (LLMs). We aimed to balance technical innovation with ethical responsibility, focusing on fairness, transparency, and minimizing bias. Efforts were made to safeguard privacy and data integrity, recognizing the potential societal impacts of our technology. We acknowledge the importance of continuous ethical evaluation and welcome constructive dialogue with the broader research community to address emerging ethical challenges in AI development.

References

Geancarlo Abich, Rafael Garibotti, Vitor Bandeira, Felipe da Rosa, Jonas Gava, Felipe Bortolon, Guilherme Medeiros, Fernando G Moraes, Ricardo Reis, and Luciano Ost. 2021. Evaluation of the soft error assessment consistency of a jit-based virtual platform simulator. *IET Computers & Digital Techniques*, 15(2):125–142.

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *Advances in Neural Information Processing Systems*, 35:17582–17596.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2021. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350.

Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Tianlong Chen, Zhenyu Zhang, Ajay Jaiswal, Shiwei Liu, and Zhangyang Wang. 2023. Sparse moe as the new dropout: Scaling dense and self-slimmable transformers. *arXiv preprint arXiv:2303.01610*.

X.AI Corp. 2024. Open release of grok-1.

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

Yong Dai, Duyu Tang, Liangxin Liu, Minghuan Tan, Cong Zhou, Jingquan Wang, Zhangyin Feng, Fan Zhang, Xueyu Hu, and Shuming Shi. 2022. One model, multiple modalities: A sparsely activated approach for text, sound, image, video and code. *arXiv preprint arXiv:2205.06126*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

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

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

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

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

Richard C Gerum, André Erpenbeck, Patrick Krauss, and Achim Schilling. 2020. Sparsity through evolutionary pruning prevents neuronal networks from overfitting. *Neural Networks*, 128:305–312.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, , and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring.

Moritz Hardt and Yu Sun. 2023. Test-time training on nearest neighbors for large language models. *arXiv preprint arXiv:2305.18466*.

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. *arXiv preprint arXiv:2310.01798*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Evgeny Kim and Roman Klinger. 2018. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*.

Josh Magnus Ludan, Qing Lyu, Yue Yang, Liam Dugan, Mark Yatskar, and Chris Callison-Burch. 2023. Interpretable-by-design text classification with iteratively generated concept bottleneck. *arXiv preprint arXiv:2310.19660*.

Lambros Malafouris. 2013. *How things shape the mind*. MIT press.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. *arXiv preprint arXiv:2305.14552*.

Masoud Monajatipoor, Mozhdeh Rouhsedaghat, Liunian Harold Li, C-C Jay Kuo, Aichi Chien, and Kai-Wei Chang. 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*, pages 725–734. Springer.

Steffen Moritz and Todd S Woodward. 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.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. 2022. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NeurIPS*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. 2023. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Gokul Subramanian Ravi, Kaitlin N Smith, Pranav Gokhale, Andrea Mari, Nathan Earnest, Ali Javadi-Abhari, and Frederic T Chong. 2021. Vaqem: A variational approach to quantum error mitigation. *arXiv e-prints*, pages arXiv–2112.

Zhen Tan, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.

Zhen Tan, Tianlong Chen, Zhenyu Zhang, and Huan Liu. 2023a. Sparsity-guided holistic explanation for llms with interpretable inference-time intervention. *arXiv preprint arXiv:2312.15033*.

Zhen Tan, Lu Cheng, Song Wang, Yuan Bo, Jundong Li, and Huan Liu. 2023b. Interpreting pretrained language models via concept bottlenecks. *arXiv preprint arXiv:2311.05014*.

LLaMA-MoE Team. 2023. Llama-moe: Building mixture-of-experts from llama with continual pre-training.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2022. Causal proxy models for concept-based model explanations. *arXiv preprint arXiv:2209.14279*.

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2017. Yedda: A lightweight collaborative text span annotation tool. *arXiv preprint arXiv:1711.03759*.

Mert Yuksekgonul, Maggie Wang, and James Zou. 2022. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, et al. 2022. Concept embedding models. In *NeurIPS 2022-36th Conference on Neural Information Processing Systems*.

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 arXiv:2205.01068*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*, 13.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. 2022a. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

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

A Definitions of Different Training Strategies

Given a text input $x \in \mathbb{R}^D$, concepts $c \in \mathbb{R}^K$ and its label y, the strategies for fine-tuning the text encoder f_θ, the projector p_ψ and the label predictor g_ϕ are defined as follows:
i) Vanilla fine-tuning an LLM: The concept labels are ignored, and then the text encoder f_θ and the label predictor g_ϕ are fine-tuned either as follows:

$$\theta, \phi = \underset{\theta,\phi}{\operatorname{argmin}} \mathcal{L}_{CE}(g_\phi(f_\theta(x), y),$$

or as follows (frozen text encoder f_θ):

$$\phi = \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{CE}(g_\phi(f_\theta(x), y),$$

where $\mathcal{L}_{CE}$ indicates the cross-entropy loss. In this work we only consider the former option for its significant better performance.
ii) Independently training LLM with the concept and task labels: The text encoder f_θ, the projector p_ψ and the label predictor g_ϕ are trained seperately with ground truth concepts labels and task labels as follows:

$$\begin{aligned} \theta, \psi &= \underset{\theta,\psi}{\operatorname{argmin}} \mathcal{L}_{CE}(p_\psi(f_\theta(x)), c), \\ \phi &= \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{CE}(g_\phi(c), y). \end{aligned}$$

During inference, the label predictor will use the output from the projector rather than the ground-truth concepts.
iii) Sequentilally training LLM with the concept and task labels: We first learn the concept encoder as the independent training strategy above, and then use its output to train the label predictor:

$$\phi = \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{CE}(g_\phi(p_\psi(f_\theta(x), y).$$

iv) Jointly training LLM with the concept and task labels: Learn the concept encoder and label predictor via a weighted sum $\mathcal{L}_{joint}$ of the two objectives described above:

$$\begin{aligned} \theta, \psi, \phi &= \underset{\theta,\psi,\phi}{\operatorname{argmin}} \mathcal{L}_{joint}(x, c, y) \\ &= \underset{\theta,\psi,\phi}{\operatorname{argmin}} [\mathcal{L}_{CE}(g_\phi(p_\psi(f_\theta(x), y) \\ &\quad + \gamma \mathcal{L}_{CE}(p_\psi(f_\theta(x)), c)]. \end{aligned}$$

It's worth noting that the LLM-CBMs trained jointly are sensitive to the loss weight γ. We tune the value for γ for better performance (Tan et al., 2023b).

For ease of reference, LLMs integrated with Concept Bottlenecks are termed LLM-CBMs (e.g., BERT-CBM). The training of LLM-CBMs is dual-faceted: (1) Ensure the concept prediction $\hat{\boldsymbol{c}} = p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ aligns with the input's true concept labels $\boldsymbol{c}$. (2) Ensure the label prediction $\hat{y} = g_{\boldsymbol{\phi}}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})))$ corresponds with true task labels y. The two objectives are *jointly* optimized, skin 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 $\mathcal{L}_{\text{joint}}$:

$$\begin{aligned} \boldsymbol{\theta}^*, \boldsymbol{\psi}^*, \boldsymbol{\phi}^* &= \underset{\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\phi}}{\operatorname{argmin}} \mathcal{L}_{\text{joint}}(\boldsymbol{x}, \boldsymbol{c}, y) \\ &= \underset{\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\phi}}{\operatorname{argmin}} [\mathcal{L}_{\text{CE}}(g_{\boldsymbol{\phi}}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) + \gamma \mathcal{L}_{\text{CE}}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})), \boldsymbol{c})] \\ &= \underset{\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\phi}}{\operatorname{argmin}} \sum_{k=1}^{K} [\mathcal{L}_{\text{CE}}(g_{\boldsymbol{\phi}_k}(p_{\boldsymbol{\psi}_k}(f_{\boldsymbol{\theta}}(x), y) + \gamma \mathcal{L}_{\text{CE}}(p_{\boldsymbol{\psi}_k}(f_{\boldsymbol{\theta}}(\boldsymbol{x})), c_k)], \end{aligned} \tag{5}$$

where, $\mathcal{L}_{\text{CE}}$ represents the Cross-Entropy loss (for regression tasks, it's replaced by the RMSE loss). The third line of the equation incorporates the loss iterating across the concepts, a detail that will prove pivotal soon. Notably, the sensitivity of jointly trained LLM-CBMs to the loss weight γ requires attention. By default, we set γ to 5.0, based on its optimized performance as observed in Tan et al. (2023b). Further details on varying training strategies are expounded in Appendix A. It should be noted that conventional LLM-CBMs (Koh et al., 2020) tend to train all concepts simultaneously. This concurrent training potentially muddles the parameters meant for individual concept prediction, thus hampering precise intervention.

B Implementation Detail

The data and implementation of our framework will be publicly released at: `https://github.com/Anonymous-submit-24/CLEAR.git`.

B.1 Experimental Setup

In this section, we provide more details on the implementation settings of our experiments. Specifically, we implement our framework with PyTorch (Paszke et al., 2017) and HuggingFace (Wolf et al., 2020) and train our framework on a single 80 GB Nvidia A100 GPU. We follow a prior work (Abraham et al., 2022) for backbone implementation. All backbone models have a maximum token number of 512 and a batch size of 8. We use the Adam optimizer to update the backbone, projector, and label predictor according to Section 3.1. The values of other hyperparameters (Table 5 in the next page) for each specific PLM type are determined through grid search. We run all the experiments on 4 Nvidia A100 GPUs with 80GB RAM.

For the LLM backbones, we use their pubic versions available on `Huggingface`. Specifically, we deploy `bert-base-uncased`, `facebook/opt-350m`, and `t5-base`. In our implementation, we also include other baseline backbones from more langugae model families. We intentionally include the above three in the main experiment results for their similar sizes. The other backbones include: `roberta-base`, `distilbert-base-uncased`, `gpt2`, `facebook/opt-125m`, `facebook/opt-1.3b`, and `switch-transformer-base`. We use logistic regression and linear regression as the head for classification and regression tasks, respectively.

Table 5: Key parameters in this paper with their annotations and evaluated values. **Bold** values indicate the optimal ones.

Notations	Specification	Definitions or Descriptions	Values
max_len	-	maximum token number of input	128 / 256 / **512**
batch_size	-	batch size	8
epoch	-	maximum training epochs	30
lr	DistilBERT	learning rate when the backbone is DistilBERT	1e-3 / 1e-4 / **1e-5** / 1e-6
	BERT	learning rate when the backbone is BERT	1e-3 / 1e-4 / **1e-5** / 1e-6
	RoBERT	learning rate when the backbone is RoBERT	1e-3 / 1e-4 / **1e-5** / 1e-6
	OPT-125M	learning rate when the backbone is OPT-125M	1e-3 / 1e-4 / **1e-5** / 1e-6
	OPT-350M	learning rate when the backbone is OPT-350	1e-4 / 1e-5 / **1e-6** / 1e-7
	OPT-1.3B	learning rate when the backbone is OPT-1.3B	1e-4 / 1e-5 / **1e-6** / 1e-7
	CLEAR	learning rate for CLEAR	1e-4 / **3e-4** / 5e-4 / 7e-4/ 1e-5
γ	DistilBERT	value of γ when the backbone is DistilBERT	1 / 3 / **5** / 7 / 9
	BERT	value of γ when the backbone is BERT	1 / 3 / **5** / 7 / 9
	RoBERT	value of γ when the backbone is RoBERT	1 / 3 / **5** / 7 / 9
	OPT-125M	value of γ when the backbone is OPT-125M	1 / 3 / **5** / 7 / 9
	OPT-350M	value of γ when the backbone is OPT-350	1 / 3 / **5** / 7 / 9
	OPT-1.3B	value of γ when the backbone is OPT-1.3B	1 / 3 / 5 / **7** / 9
	CLEAR	value of γ for CLEAR	5 / 7 / 9 / **10** / 11 / 13 / 15

C Description of Datasets

In this section, we provide detailed descriptions of the benchmark datasets used in our experiments. Thier specific concepts are presented in Table 1.

- `CEBaB` (Abraham et al., 2022) contains restaurant reviews from Opentable. Possible labels include 1 Star, 2 Stars, 3 Stars, 4 Stars, 5 Stars, indicating different sentiment score with 5 Stars indicating the most positive sentiment.
- `IMDB-C` (Tan et al., 2023b) consists of movie reviews from IMDB datasets. Possible labels include positive and negative.
- `ASAP-C` is comprised of students essays with their scores from the ASAP dataset (Hamner et al., 2012). The original scores range from 0 - 100. In our study, we evenly split the datasets into 10 grade categories, ranging from 0 - 9, corresponding to 10 widely-used letter grades, D, C-, C, C+, ..., A, A+. We know that in real-world, students' grades tend to be normally distributed. Here we use even split to make the task easier by mitiagting the class imbalance issue, which is out of the scope of this work.

C.1 Data Anotation for `ASAP-C`

Our annotation policy is following a previous work (Cai et al., 2021) for NLP datasets annotating. For the `ASAP` dataset, we annotate the four concepts (Contents, Reasoning, Language, Supportiveness) manually. Even though the concepts are naturally understandable by humans, two Master students familiar with English writing tutoring are selected as annotators for independent annotation with the annotation tool introduced by Yang et al. (2017). The strict quadruple matching F1 score between two annotators is 87.3%, which indicates a consistent agreement between the two annotators (Kim and Klinger, 2018). In case of disagreement, a third expert will be asked to make the final decision. The instruction is as follows (the concepts are listed in Table 1):

a. According to the essay "$\{text_1\}$", the "$\{concept_1\}$" of the essay is "positive".
b. According to the essay "$\{text_2\}$", the "$\{concept_2\}$" of the essay is "negative".
c. According to the essay "$\{text_3\}$", the "$\{concept_3\}$" of the essay is "unknown".
d. According to the essay "$\{text_i\}$", how is the "$\{concept_i\}$" of the essay? Please answer with one option in "positive, negative, or unknown".

D Acknowledgment of AI Assistance in Writing and Revision

We utilized ChatGPT-4 for revising and enhancing sections of this paper.

E Comparative Results on the ASAP-C dataset

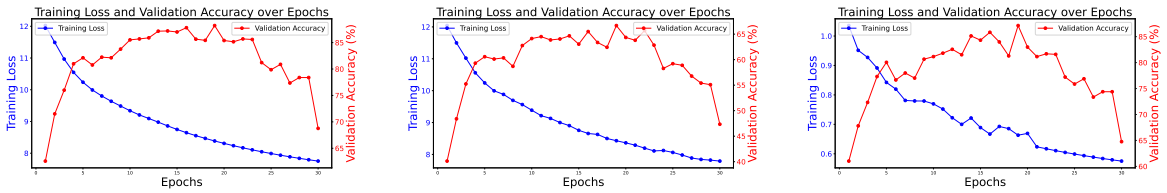
Table 6: Comparative results on the ASAP-C dataset, using *Macro F1* (↑) as the evaluation metric for concept classification, expressed in percentages (%) and *RMSE* (↓) as the evaluation metric for essay score regression. Scores shaded in gray highlight instances where the model experienced catastrophic 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.

Methods	Backbones	ASAP-C							
		Pre-intervention				Post-intervention			
		Dev		Test		Dev		Test	
		Concept (F1 ↑)	Task (MSE ↓)	Concept (F1 ↑)	Task (MSE ↓)	Concept (F1 ↑)	Task (MSE ↓)	Concept (F1 ↑)	Task (MSE ↓)
Direct Intervention Methods									
Prompting	GPT4	-	1.637	-	1.534	-	1.637	-	1.685
Fine-tuning	BERT	-	0.804	-	0.753	-	0.939	-	0.626
	OPT	-	0.769	-	0.728	-	0.862	-	0.604
	T5	-	0.752	-	0.714	-	0.842	-	0.581
ITI	T5	-	0.752	-	0.714	-	0.752		0.634
Concept Bottleneck Models									
Vanilla-CBMs	BERT	81.24	0.896	80.67	0.904	81.24	0.896	83.68	0.884
	OPT	83.62	0.853	82.64	0.872	83.62	0.853	84.24	0.842
	T5	85.34	0.834	84.36	0.857	85.34	0.834	86.69	0.826
LF-CBMs	BERT	77.64	1.034	76.48	1.165	77.64	1.034	77.96	0.980
	OPT	78.57	0.924	77.26	0.968	78.57	0.924	76.18	1.158
	T5	79.66	0.864	78.81	0.891	79.66	0.864	78.48	0.936
CEMs	BERT	82.37	0.867	82.64	0.856	82.37	0.867	83.79	0.796
	OPT	84.41	0.842	83.29	0.879	84.41	0.842	86.67	0.723
	T5	86.58	0.704	85.62	0.713	86.58	0.704	88.32	0.684
Metacognition Intervention									
CLEAR	OPT-MoCE	85.63	0.765	85.27	0.771	85.63	0.765	88.24	0.679
CLEAR	T5-MoCE	87.62	0.684	87.35	0.694	87.62	0.684	89.65	0.624

F Comparison with Existing Works on MoE for LLMs

Mixture of Experts in Large Language Models. The incorporation of Mixture of Experts (MoE) into Large Language Models (LLMs) has evolved significantly, with early research by Shazeer et al. (2017) laying the groundwork. These foundational studies (Fedus et al., 2022; Zhou et al., 2022a; Du et al., 2022; Artetxe et al., 2021; Shen et al., 2023) focused primarily on improving model performance and computational efficiency in a black-box manner. On the contrary, in this work, we utilize the design of MoE in LLMs for metacognitive capabilities. This novel approach, distinct from earlier efficiency-focused applications, uses MoE for error detection and correction, a critical step towards solving the interpretability and trust issues in AI decision-making. Our framework, CLEAR, contributes to this evolving landscape by embedding MoE within a metacognitive framework, emphasizing error rectification, transparency, and autonomy in LLMs. This shift marks a significant advancement from traditional MoE applications, positioning CLEAR at the forefront of innovative LLM enhancement strategies.

G Analysis of Overfitting in Concept Learning



(a) CEBaB (b) IMDB-C (c) ASAP-C

Figure 7: Visualization of training dynamics of one run on CEBaB, IMDB-C and ASAP-C datasets. We adopt the "early stop" strategy to avoid overfitting, where models with the highest validation accuracy are selected and evaluated on test sets.

H More Examples from Real-world Datasets

This place is super cool. Felt like I was in NYC vs downtown Phoenix They have hip hop playing and a cool staff They offer something for everyone Ice cream, coffee, beer, wine, drinks, food, whatever you want. The beer selection is actually better than most bars I've been too and high end joints. Old Rasputin on nitro. What the pho? Great choice. I'd come back for sure and highly recommend!

Y	Service	Food	Ambiance	Noisy	Task Label
c_k/y	+	+	+	Unk	4
$\hat{c}_k/\hat{y}$	+	+	+	+	5
$\hat{c}'_k/\hat{y}'$	+	+	+	Unk	4

Figure 8: An example for the metacognitive intervention on one instance from the CEBaB dataset.

Some films just simply should not be remade. This is one of them. In and of itself it is not a bad film. But it fails to capture the flavor and the terror of the 1963 film of the same title. Liam Neeson was excellent as he always is, and most of the cast holds up, with the exception of Owen Wilson, who just did not bring the right feel to the character of Luke. But the major fault with this version is that it strayed too far from the Shirley Jackson story in it's attempts to be grandiose and lost some of the thrill of the earlier film in a trade off for snazzier special effects. Again I will say that in and of itself it is not a bad film. But you will enjoy the friction of terror in the older version much more.

	Emotional Arousal	Acting	Storyline	Cinemato-graphy	Task Label
c_k/y	-	+	Unk	-	+
$\hat{c}_k/\hat{y}$	-	-	Unk	-	-
$\hat{c}'_k/\hat{y}'$	-	+	Unk	-	+

Figure 9: An example for the metacognitive intervention on one instance from the IMDB-C dataset.

I am not a patient person at all. But sometimes I have to be like my birthday for instance, I would love it if my birthday came at least every month. But of course, I only have @NUM1 birthday a year, so I have to wait. I would like to be a patient person. It's just not in the cards for me. My father on the other hand is more patient than anyone. I know he will tell me to clean the car @DATE1 I told him I didn't do it yet so he says he will give me more time. I couldn't be that patient with my kids. I would tell them to clean it now or they would be grounded. I wouldn't force them to or anything but I'm not gonna wait a whole month before I get my car cleaned! I guess I could try to be as patient with my father but that would be really hard. Although if I'm "patient," I'm sure I will be able to do it!

Y	Content	Reasoning	Language	Supportive-ness	Task Label
c_k/y	+	+	-	-	6
$\hat{c}_k/\hat{y}$	+	+	+	Unk	8
$\hat{c}'_k/\hat{y}'$	+	+	-	-	6

Figure 10: An example for the metacognitive intervention on one instance from the ASAP-C dataset.