Sparsity-Guided Holistic Explanation for LLMs with Interpretable Inference-Time Intervention

Zhen Tan[1], Tianlong Chen[2], Zhenyu Zhang[3], Huan Liu[1]

[1]Arizona State University
[2]University of North Carolina at Chapel Hill
[3]University of Texas at Austin
{ztan36,huanliu}@asu.edu, tianlong@cs.unc.edu, zhenyu.zhang@utexas.edu

Abstract

Large Language Models (LLMs) have achieved unprecedented breakthroughs in various natural language processing domains. However, the enigmatic "black-box" nature of LLMs remains a significant challenge for interpretability, hampering transparent and accountable applications. While past approaches, such as attention visualization, pivotal subnetwork extraction, and concept-based analyses, offer some insight, they often focus on either local or global explanations within a single dimension, occasionally falling short in providing comprehensive clarity. In response, we propose a novel methodology anchored in sparsity-guided techniques, aiming to provide a holistic interpretation of LLMs. Our framework, termed *SparseCBM*, innovatively integrates sparsity to elucidate three intertwined layers of interpretation: input, subnetwork, and concept levels. In addition, the newly introduced dimension of interpretable inference-time intervention facilitates dynamic adjustments to the model during deployment. Through rigorous empirical evaluations on real-world datasets, we demonstrate that SparseCBM delivers a profound understanding of LLM behaviors, setting it apart in both interpreting and ameliorating model inaccuracies. Codes are provided in supplements.

Introduction

The advent of Large Language Models (LLMs) has captured the intricacies of language patterns with striking finesse, rivaling, and at times, surpassing human performance (Zhou et al. 2022; OpenAI 2023). However, their laudable success story is shadowed by a pressing concern: a distinct lack of *transparency* and *interpretability*. As LLMs burgeon in complexity and scale, the elucidation of their internal mechanisms and decision-making processes has become a daunting challenge. The opaque "black-box" characteristics of these models obfuscate the transformation process from input data to generated output, presenting a formidable barrier to trust, debugging, and optimal utilization of these potent computational tools. Consequently, advancing the interpretability of LLMs has emerged as a crucial frontier in machine learning and natural language processing research, aiming to reconcile the dichotomy between superior model performance and comprehensive usability.
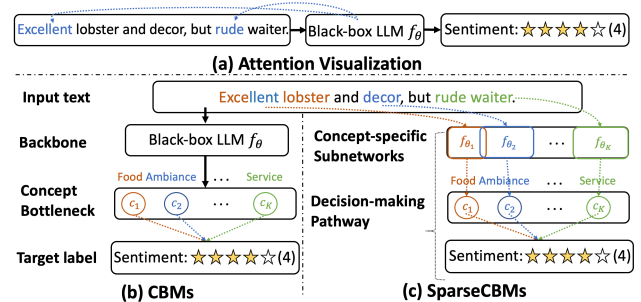
Figure 1: The illustration includes: (a) *Attention visualization* provides a localized, attention-driven explanation. While insightful, this might be less decipherable or intuitive for users outside the realm of computer science. (b) *CBMs* deliver a broader, concept-level understanding, resonating naturally with human cognition. However, they sometimes miss out on the nuanced, granular insights of the LLM's workings. (c) *SparseCBMs* outline a holistic decision pathway for each input, seamlessly progressing from tokens, via pertinent subnetworks and concepts, to the final task label. This approach marries the strengths of both local and global explanations, addressing their respective shortcomings.

The spectrum of interpretability solutions for language models can be broadly bifurcated into two categories. ❶ Initial approaches predominantly leverage *local explanations*, employing techniques such as visualization of attention weights (Galassi, Lippi, and Torroni 2020), probing of feature representations (Mishra, Sturm, and Dixon 2017; Lundberg and Lee 2017), and utilization of counterfactuals (Wu et al. 2021; Ross, Marasović, and Peters 2021), among others. These methods focus on providing explanations at granular levels, such as individual tokens, instances, neurons, or subnetworks, as exemplified in Figure 1 (a). While these low-level explanations offer a degree of reliability, they often sacrifice **readability** and **intuitiveness** (Losch, Fritz, and Schiele 2019), thereby constraining their practical applicability. ❷ More recently, researchers have tended to *global explanations*, such as concept-based analyses that inherently resonate with human cognition (Wang et al. 2023a; Abraham et al. 2022). For instance, one recent work (Tan et al. 2023) incorporates Concept Bottleneck Models (CBMs) (Koh et al.

2020) into pretrained language models, leading to an impressive "interpretability-utility" Pareto front. Figure 1 (b) exemplifies this for sentiment analysis tasks, where human-intelligible concepts like "Food", "Ambiance", and "Service" correspond to neurons in the concept bottleneck layer. The final decision layer is designed as a linear function of these concepts, rendering the decision rules easily understandable. However, these methods excessively focus on global explanations. The **underlying reasoning** between raw input and concepts remains unclear.

To address these limitations, our work champions a *holistic* interpretation of LLM predictions. We unveil *SparseCBM*, an evolved CBM variant that melds the complementary "strengths" of local and global explanations, thereby addressing the individual "weaknesses" of each. This confluence is born from rigorous sparsity-guided refinement designed specifically for LLMs, as depicted in Figure 1 (c). Concretely, SparseCBM iteratively prunes the LLM backbone guided by a joint objective of optimizing for both concept and task labels until the desired sparsity level is accomplished. This exercise distills the LLM into distinct yet interconnected subnetworks, each corresponding to a predefined concept. As such, SparseCBM provides a comprehensive and intelligible decision-making pathway for each input text, tracing from tokens through subnetworks and concepts, ultimately leading to the final task label.

Another unique feature is that, SparseCBMs allow **interpretable inference-time intervention** (Koh et al. 2020; Li et al. 2023a). The inherent sparsity-driven structure of SparseCBM allows it to adjust its internal parameters dynamically, based on the context of the input. In practical terms, this means that, during inference, SparseCBM can identify potential areas of ambiguity or misconception, and proactively modify its internal decision-making routes without a full-scale retraining. This "on-the-fly" adaptability not only enhances prediction accuracy but also offers users a window into how the model adjusts its reasoning in real time. By making these modifications both accessible and understandable, SparseCBM bridges the common chasm between interpretability and agility for LLMs. This real-time decision pathway modification, stands as a beacon for fostering trust and facilitating more nuanced human-model interactions. In summary, SparseCBM carries the following advantages:

- *Empirical Validation*: Our experiments reveal that SparseCBM enables interpretability at the token, subnetwork, and concept levels, creating a synergy that surpasses the mere aggregation of these elements.

- *Superior Performance*: SparseCBM demonstrates state-of-the-art performance on conventional benchmarks, both in terms of concept and task label predictions.

- *Metacognitive Inference-Time Intervention*: Compared to vanilla CBMs, SparseCBM exhibits a unique capability for efficient and interpretable inference-time intervention. By subtly modulating internal sparsity, SparseCBM learns to sidestep known pitfalls. This property bolsters user trust in SparseCBMs and, by extension, LLMs.

Related Work

Interpreting Language Models

Research on the interpretability of language models has been robust, with previous work focusing on visualization of hidden states and attention weights in transformer-based models (Vig 2019; Galassi, Lippi, and Torroni 2020). These techniques, while valuable, often provided granular insights that were not easily interpretable at a high level. Feature importance methods like LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017) provided valuable insights into how each input feature contributes to the prediction, but still fail to offer a global understanding of the model behavior, and often lack intuitiveness and readability.

The advent of concept-based interpretability has marked a significant development, offering more global, high-level explanations (Koh et al. 2020; Abraham et al. 2022; Wang et al. 2023a). Concept Bottleneck Models (CBMs) (Koh et al. 2020; Oikarinen et al. 2023) which incorporate a concept layer into the model, have gained traction recently (Tan et al. 2023). CBMs are trained with task labels and concept labels either *independently*, *sequentially*, or *jointly*. This design enables inference-time debugging by calibrating the activations of concepts. Yet, current CBMs are deficient in their ability to offer granular interpretations, and inference-time interventions remain incapable of altering the language model backbone, leading to recurrent errors. On the other hand, the interpretability of LLMs remains a less explored area. Although some progress has been made, such as guiding LLMs to generate explanations for their predictions using finely tuned prompts (Li et al. 2022), the reliability of these explanations remains questionable. In summary, a reliable method facilitating holistic insights into model behavior is still wanting. In response, our work advances this field by introducing SparseCBM, a holistic interpretation framework for LLMs that tackles both local and global interpretations, thus enhancing the usability and trustworthiness of LLMs.

Sparsity Mining for Language Models

Sparsity-driven techniques, often associated with model pruning, form an energetic subset of research primarily in the pursuit of model compression. At their core, these methods focus on the elimination of less influential neurons while retaining the more critical ones, thereby sustaining optimal model performance (LeCun, Denker, and Solla 1990a; Han, Mao, and Dally 2016; Han et al. 2015; LeCun, Denker, and Solla 1990b; Liu et al. 2017; He, Zhang, and Sun 2017; Zhou, Alvarez, and Porikli 2016). Contemporary research has shed light on the heightened robustness of pruned models against adversarial conditions, such as overfitting and distribution shifts. Typical pruning methods for language models encompass structured pruning (Michel, Levy, and Neubig 2019), fine-grained structured pruning (Lagunas et al. 2021), and unstructured pruning (Gale, Elsen, and Hooker 2019). In brief, unstructured pruning removes individual weights in a network, leading to a sparse matrix, structured pruning eliminates entire structures like neurons or layers for a dense model, while fine-grained structured pruning prunes smaller structures like channels or weight

vectors, offering a balance between the previous two. We direct the readers to the benchmark (Liu et al. 2023) for a comprehensive overview. In our case, we focus on unstructured pruning for its effectiveness and better interpretability.

Recently, studies have underscored the interpretability afforded by sparse networks (Subramanian et al. 2018). For instance, Meister et al. (2021) delve into the interpretability of sparse attention mechanisms in language models, Liu et al. (2022) incorporate sparse contrastive learning in an ancillary sparse coding layer to facilitate word-level interpretability, and Oikarinen et al. (2023) demonstrate that a sparsity constraint on the final linear predictor enhances concept-level interpretation of CBMs. Despite their effectiveness, these frameworks restrict sparsity to a handful of layers, leading to unidimensional interpretability that falls short of the desired comprehensiveness. In contrast, our proposed framework, SparseCBM, imposes sparsity across the entire LLM backbone, enabling holistic interpretation at the token, subnetwork, and concept levels.

Methodology

Preliminary: Concept Bottleneck Models for Language Models

Problem Setup. In this study, we aim to interpret the predictions of fine-tuned Large Language Models (LLMs) in text classification tasks. Given a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{c}^{(i)})_{i=1}^{N}\}$, we consider an LLM $f_{\boldsymbol{\theta}}$ that encodes an input text $\boldsymbol{x} \in \mathbb{R}^{D}$ into a latent representation $\boldsymbol{z} \in \mathbb{R}^{E}$, and a linear classifier $g_{\boldsymbol{\phi}}$ that maps $\boldsymbol{z}$ into the task label y.

Incorporate Concept Bottlenecks for Large Language Models. Our architecture mainly follows Tan et al. (2023). Instead of modifying LLM encoders, which could significantly affect the quality of the learned text representation, we introduce a linear layer with sigmoid activation $p_{\boldsymbol{\psi}}$. This layer projects the learned latent representation $\boldsymbol{z} \in \mathbb{R}^{E}$ into the concept space $\boldsymbol{c} \in \mathbb{R}^{K}$, resulting in a pathway represented as $\boldsymbol{x} \rightarrow \boldsymbol{z} \rightarrow \boldsymbol{c} \rightarrow y$. Here, we allow multiclass concepts for more flexible interpretation. For convenience, we represent CBM-incorporated LLMs as LLM-CBMs (e.g., BERT-CBM). LLM-CBMs are trained with two objectives: (1) align concept prediction $\hat{\boldsymbol{c}} = p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ to $\boldsymbol{x}$'s ground-truth concept labels $\boldsymbol{c}$, and (2) align label prediction $\hat{y} = g_{\boldsymbol{\phi}}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})))$ to ground-truth task labels y. We mainly experiment with our framework optimized through the *joint training* strategy for its significantly better performance, as also demonstrated in Tan et al. (2023). Jointly training LLM with the concept and task labels entails learning the concept encoder and label predictor via a weighted sum, $\mathcal{L}_{joint}$, of the two objectives:

$$\boldsymbol{\theta}^{*}, \boldsymbol{\psi}^{*}, \boldsymbol{\phi}^{*} = \underset{\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}}{\arg\min} \mathcal{L}_{joint}(\boldsymbol{x}, \boldsymbol{c}, y)$$
$$= \underset{\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}}{\arg\min}[\mathcal{L}_{CE}(g_{\boldsymbol{\phi}}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})), y) \quad (1)$$
$$+ \gamma \mathcal{L}_{CE}(p_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})), \boldsymbol{c})].$$

It's worth noting that the LLM-CBMs trained jointly are sensitive to the loss weight γ. We set the default value for γ as

5.0 for its better performance (Tan et al. 2023). Despite the promising progress having been made, present LLM-CBMs typically train all concepts concurrently, leading to intertwined parameters for concept prediction, making the process less transparent and hampering targeted intervention.

SparseCBMs

To address the aforementioned issue, the goal of this paper is to provide a holistic and intelligible decision-making pathway for each input text, tracing from tokens through subnetworks and concepts, ultimately leading to the final task label. To this end, we introduce *SparseCBM*, a pioneering framework capable of unraveling the intricate LLM architectures into a number of concept-specific subnetworks. Our approach not only outperforms conventional CBMs in concept and task label prediction performance but also proffers enhanced interpretation concerning neuron activations, for instance, illuminating which weights inside the LLM backbone play pivotal roles in learning specific concepts.

Our framework starts with decomposing the joint optimization defined in Eq. (1) according to each concept c_k, which is formulated as follows:

$$\boldsymbol{\theta}^{*}, \boldsymbol{\psi}^{*}, \boldsymbol{\phi}^{*} = \{(\boldsymbol{\theta}_{k}^{*})_{k=1}^{K}\}, \{(\boldsymbol{\psi}_{k}^{*})_{k=1}^{K}\}, \{(\boldsymbol{\phi}_{k}^{*})_{k=1}^{K}\}$$
$$= \underset{\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}}{\arg\min} \sum_{k=1}^{K} \mathcal{L}_{joint}(\boldsymbol{x}, c_{k}, y).$$
$$\quad (2)$$
$$= \underset{\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}}{\arg\min} \sum_{k=1}^{K} [\mathcal{L}_{CE}(g_{\boldsymbol{\phi}_{k}}(p_{\boldsymbol{\psi}_{k}}(f_{\boldsymbol{\theta}}(x)), y)$$
$$+ \gamma \mathcal{L}_{CE}(p_{\boldsymbol{\psi}_{k}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})), c_{k})],$$

where $\boldsymbol{\phi}_k, \boldsymbol{\psi}_k$ are the weights of the kth parameter of the projector and classifier, and $\boldsymbol{\theta}_k$ is the subnetwork specific for the concept c_k, which is explained later. Since both of them are comprised of a single linear layer (with or without the activation function), the involved parameters for c_k can be directly indexed from these models and are self-interpretable (Koh et al. 2020; Tan et al. 2023).

The remaining task is to excavate concept-specific subnetworks for each concept from the vast architecture of Large Language Models (LLMs). The guiding intuition behind this strategy is to perceive the prediction of concept labels as individual classification tasks, ones that should not strain the entirety of pretrained LLMs given their colossal reserves of knowledge encapsulated in multi-million to multi-billion parameters. We propose an unstructured pruning of the LLM backbone for each concept classification task, such that distinct pruned subnetworks are accountable for different concepts while preserving prediction performance.

Holistic and Intelligible Decision-making Pathways. We leverage unstructured pruning strategies to carve out concept-specific subnetworks within the LLM backbones. The noteworthy edge of such unstructured pruning strategies lies in their ability to engender weight masks in accordance with the weight importance. Such masks naturally can offer an immediate and clear interpretation. Concretely, we introduce a 0/1 weight mask M_k for each corresponding subnetwork. Consequently, the weights of each subnetwork can be

represented as $\boldsymbol{\theta}_{M_k} = \boldsymbol{M}_k \odot \boldsymbol{\theta}^*$, representing the Hadamard (element-wise) product between the LLM weights $\boldsymbol{\theta}^* \in \mathbb{R}^L$ and the weight mask $\boldsymbol{M}_k \in \{0,1\}^L$ for the concept c_k.

With well-optimized $\{(\boldsymbol{M})_{k=1}^K\}$, during inference, the decision-making pathway can be represented as:

$$\hat{y} = \sum_{k=1}^K \boldsymbol{\phi}_k^* \cdot \sigma(\boldsymbol{\psi}_k^* \cdot f_{\boldsymbol{\theta}_{M_k}}(\boldsymbol{x})) = \sum_{k=1}^K \boldsymbol{\phi}_k^* \cdot \sigma(\boldsymbol{\psi}_k^* \cdot f_{\boldsymbol{M}_k^\odot \boldsymbol{\theta}^*}(\boldsymbol{x})),$$
(3)

where $\sigma(\cdot)$ is the sigmoid activation function of the projector. This decision-making pathway defined in Eq. (3) factorizes the parameters of the SparseCBM, and can be optimized through one backward pass of the discomposed joint loss defined in Eq. (2) with $\boldsymbol{\theta}_k^* = \boldsymbol{\theta}_{M_k}$. Importantly, we posit that such decision-making pathways can deliver holistic explanations for the model's predictions. For instance, by scrutinizing the weights in the classifier g_ϕ and the concept activation post the σ function, we can get a concept-level explanation regarding the importance of different concepts. Also, visualizing each subnetwork mask $\boldsymbol{M}_k$ will furnish a subnetwork-level comprehension of neuron behavior and its importance in acquiring a specific concept and forming predictions. Additionally, the study of the gradient of input tokens in masked concept-specific subnetworks can provide more accurate token-concept mapping. Notably, our experiments demonstrate that SparseCBMs, in addition to providing multi-dimensional interpretations, can match or even surpass their dense counterparts in performance on both concept and task label prediction. Another unique feature of SparseCBMs lies in that, the weight masks $\{(\boldsymbol{M}_k)_{k=1}^K\}$ engendered by unstructured pruning facilitates the process of efficient and interpretable *Sparsity-based Inference-time Intervention*, which is expounded later.

Concept-Induced Sparsity Mining. Next, we elaborate on how to compute those sparsity masks, given an optimized LLM backbone. A second-order unstructured pruning (Hassibi and Stork 1992; Kurtic et al. 2022) for LLMs has been incorporated. Initially, the joint loss $\mathcal{L}$ (we omit the subscript $joint$ for brevity in subsequent equations) can be expanded at the weights of subnetwork $\boldsymbol{\theta}_{M_k}$ via Taylor expansion:

$$\mathcal{L}(\boldsymbol{\theta}_{M_k}) \simeq \mathcal{L}(\boldsymbol{\theta}^*) + (\boldsymbol{\theta}_{M_k} - \boldsymbol{\theta}^*)^\top \nabla \mathcal{L}(\boldsymbol{\theta}^*)$$
$$+ \frac{1}{2}(\boldsymbol{\theta}_{M_k} - \boldsymbol{\theta}^*)^\top \boldsymbol{H}_{\mathcal{L}}(\boldsymbol{\theta}^*)(\boldsymbol{\theta}_{M_k} - \boldsymbol{\theta}^*),$$
(4)

where $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{\theta}^*)$ stands for the Hessian matrix of the decomposed joint loss at $\boldsymbol{\theta}^*$. Since $\boldsymbol{\theta}^*$ is well-optimized, we assume $\nabla \mathcal{L}(\boldsymbol{\theta}^*) \approx 0$ as the common practice (Hassibi and Stork 1992; Kurtic et al. 2022). Then, the change in loss after pruning can be represented as:

$$\Delta \mathcal{L}(\Delta \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}_{M_k}) - \mathcal{L}(\boldsymbol{\theta}^*) \simeq \frac{1}{2}\Delta \boldsymbol{\theta}^\top \boldsymbol{H}_{\mathcal{L}}\Delta \boldsymbol{\theta}, \quad (5)$$

where, $\Delta \boldsymbol{\theta} = \boldsymbol{\theta}_{M_k} - \boldsymbol{\theta}^*$ signifies the change in LLM weights, that is, pruned parameters. Given a target sparsity $s \in [0, 1)$, we seek the minimum loss change incurred by pruning. In our case, the default sparsity is designed as: $s \geq 1 - \frac{1}{K}$, implying each subnetwork contains a maximum of $\frac{1}{K}$ parameters in the dense counterpart. Ideally, we desire

separate parameters in the LLM backbone to ensure optimal interpretability. Then, the problem of computing the sparsity masks can be formulated as a constrained optimization task:

$$\min_{\Delta \boldsymbol{\theta}} \quad \frac{1}{2}\Delta \boldsymbol{\theta}^\top \boldsymbol{H}_{\mathcal{L}}(\boldsymbol{\theta}^*)\Delta \boldsymbol{\theta},$$
$$s.t. \quad \boldsymbol{e}_b^\top \Delta \boldsymbol{\theta} + \theta_b = 0, \quad \forall b \in \boldsymbol{Q},$$
(6)

where $\boldsymbol{e}_b$ denotes the bth canonical basis vector of the block of weights $\boldsymbol{Q}$ to be pruned. This optimization can be solved by approximating the Hessian at $\boldsymbol{\theta}^*$ via the dampened empirical Fisher information matrix (Hassibi and Stork 1992; Kurtic et al. 2022). Hence, we can derive the optimized concept-specific masks $\{(\boldsymbol{M}_k)_{k=1}^K\}$. More details are in Appendix C.

Sparsity-based Inference-time intervention. SparseCBMs also exhibit the capability to allow inference-time concept intervention (a trait inherited from CBMs), thus enabling more comprehensive and user-friendly interactions. SparseCBMs allow modulation of the inferred concept activations: $\hat{\boldsymbol{a}} = \sigma(p_\phi(f_{\boldsymbol{\theta}}(\boldsymbol{x})))$. There are two straightforward strategies for undertaking such intervention. The first option is the *oracle intervention* (Koh et al. 2020), where human experts manually calibrate the concept activations $\hat{\boldsymbol{a}}$ and feed them into the classifier. Despite its apparent simplicity, oracle intervention directly operates on concept activations and, therefore, cannot fix the flawed mapping learned by the LLM backbone. As a consequence, the model will replicate the same error when presented with the same input. Meanwhile, another strategy involves further fine-tuning the LLM backbone on the test data. However, this approach is not only inefficient but also has a high risk of leading to significant overfitting on the test data. Those limitations present a barrier to the practical implementation of CBMs in high-stakes or time-sensitive applications.

As a remedy, we further propose a *sparsity-based intervention* that is self-interpretable and congruent with SparseCBMs. It helps LLMs to learn from each erroneously predicted concept during inference time, while preserving overall performance. The core idea is to subtly modify the concept-specific masks for the LLM backbone when a mispredicted concept is detected. Specifically, parameters of the LLM backbone f_θ, projector p_ψ, and the classifier g_ϕ are frozen, while the concept-specific masks $\{(\boldsymbol{M}_k)_{k=1}^K\}$ is kept trainable. During the test phase, if a concept prediction $\hat{c}_k$ for an input text $\boldsymbol{x}$ is incorrect, we acquire the gradient $\mathcal{G}_k(\boldsymbol{x})$ for the corresponding subnetwork $f_{\boldsymbol{\theta}_{M_k}}$, and modulate the learned mask $\boldsymbol{M}_k$ accordingly.

Inspired by Evci et al. (2020); Sun et al. (2023), we define the saliency scores for LLM parameters by the l_2-norm of the product of the gradient of the mask and the parameter weights: $\mathcal{S} = \|\mathcal{G}_k(\boldsymbol{x}) \cdot \boldsymbol{\theta}^*\|$. Subsequently, we perform the following two operations based on the saliency scores: (1) **Drop** a proportion of r unpruned weights with the lowest saliency scores: $\arg\min_m^{r \cdot |\boldsymbol{\theta}|} \mathcal{S}_m, \forall m \in |\boldsymbol{\theta}_{M_k}|$. (2) **Grow** a proportion of r pruned weights with the highest saliency scores: $\arg\max_m^{r \cdot |\boldsymbol{\theta}|} \mathcal{S}_m, \forall m \in |\boldsymbol{\theta} \setminus \boldsymbol{\theta}_{M_k}|$. Here m refers to the parameter index of the LLM backbone. By dropping and growing an equal number of parameters, the overall spar-

Dataset	CEBaB (5-way classification)			
	Train/Dev/Test		1755 / 1673 / 1685	
	Concept	**Negative**	**Positive**	**Unknown**
Concept	Food	1693 (33%)	2087 (41%)	1333 (26%)
	Ambiance	787 (15%)	994 (20%)	3332 (65%)
	Service	1249 (24%)	1397 (27%)	2467 (49%)
	Noise	645 (13%)	442 (9%)	4026 (78%)
Dataset	IMDB-C (2-way classification)			
	Train/Dev/Test		100 / 50 / 50	
	Concept	**Negative**	**Positive**	**Unknown**
Concept	Acting	76 (38%)	66 (33%)	58 (29%)
	Storyline	80 (40%)	77 (38%)	43 (22%)
	Emotion	74 (37%)	73 (36%)	53 (27%)
	Cinematography	118 (59%)	43 (22%)	39 (19%)

Table 1: Statistics of experimented datasets and concepts.

sity s of the LLM backbone remains unchanged. This mask-level intervention is further optimized through the decomposed joint loss $\mathcal{L}_{joint}$ defined in Eq. (2). Note that r is set as a relatively small value (e.g., 0.01) to compel the model to retain the overall performance while learning from the mistake. Our experiments validate that the proposed sparsity-based intervention can effectively enhance inference-time accuracy without necessitating training of the entire LLM backbone. Also, the intervened parameters provide insight into the parameters that contributed to each misprediction.

Experiments

Experimental Setup

Datasets. Our experiments are conducted on two widely-used real-world datasets: CEBaB (Abraham et al. 2022) and IMDB-C (Tan et al. 2023). Each of them is a text-classification dataset comprised of human-annotated concept and task labels. Their statistics are presented in Table 1.

LLM backbones. In this research, we primarily consider two widely-recognized, open-source lineages of pretrained LLMs: the BERT-family models (Devlin et al. 2018; Liu et al. 2019; Sanh et al. 2019) and OPT-family models (Zhang et al. 2022). Specially, we also include directly prompting GPT4 (OpenAI 2023) as a baseline to let it generate concept and task labels for given texts. Even though being proprietary, GPT4 is widely regarded as the most capable LLM currently, so we choose it as the baseline backbone. For better performance, we obtain the representations of the input texts by pooling the embedding of all tokens. Reported scores are the averages of three independent runs. Our work is based on general text classification implementations. The PyTorch Implementation is available at *https://github.com/Zhen-Tan-dmml/SparseCBM.git*.

Interpretability

Utility v.s. Interpretability. Table 2 presents the performance of the concept and task label prediction:

Backbone	Acc. / F1	CEBaB		IMDB-C	
		Concept	**Task**	**Concept**	**Task**
GPT4	Prompt	75.9/71.5	51.3/45.9	64.5/61.5	71.4/68.7
DistilBERT	Standard	-	70.3/80.4	-	77.1/73.8
	CBM	81.1/83.5	63.9/76.5	67.5/63.8	76.5/69.8
	SparseCBM	**82.0/84.0**	**64.7/77.1**	**68.4/64.3**	**76.9/71.4**
BERT	Standard	-	67.9/79.8	-	78.3/72.1
	CBM	83.2/85.3	66.9/78.1	68.2/62.8	77.3/70.4
	SparseCBM	**83.5/85.6**	66.9/**79.1**	**69.8/65.2**	76.5/**71.6**
RoBERTa	Standard	-	71.8/81.3	-	82.2/77.3
	CBM	82.6/84.9	70.1/81.3	69.9/68.9	81.4/79.3
	SparseCBM	**82.8/85.5**	**70.3/81.4**	**70.2/69.7**	**81.5/79.9**
OPT-125M	Standard	-	70.8/81.4	-	84.3/80.0
	CBM	85.4/87.3	68.9/79.7	68.7/66.5	81.8/78.2
	SparseCBM	**86.2/88.0**	68.9/**79.8**	**70.0/67.4**	**82.6/79.9**
OPT-350M	Standard	-	71.6/82.6	-	86.4/83.5
	CBM	87.8/89.4	69.9/80.7	72.6/70.5	84.5/82.4
	SparseCBM	87.3/88.7	68.2/79.8	**73.3/71.1**	**85.0/82.5**
OPT-1.3B	Standard	-	74.7/83.9	-	88.4/83.7
	CBM	90.0/91.5	73.6/82.1	76.8/74.6	85.7/83.3
	SparseCBM	89.9/**91.6**	**73.8/82.6**	76.4/**74.7**	**86.6/83.9**

Table 2: Comparisons of task accuracy and interpretability using CEBaB and IMDB-C datasets with BERT-family and OPT-family models as the backbones. Metrics for both task and concept labels are *Accuracy/Macro F1* in %. A score in **bold** indicate that the SparseCBM under the current setting outperforms its dense CBM counterpart.

- **Multidimensional Interpretability:** SparseCBMs stand out by offering multidimensional interpretability without compromising task prediction performance. In comparison with standard LLMs (which are fine-tuned exclusively with task labels), SparseCBMs grant concept-level interpretability with only a slight dip in task prediction accuracy. Impressively, SparseCBMs can match or even outperform their dense CBM counterparts while providing multifaceted explanations that extend beyond mere concepts. This underlines the potency of SparseCBMs in striking a balance between interpretability and utility.

- **Scalability with Larger LLM Backbones:** The utilization of larger LLMs within SparseCBMs leads to superior interpretability-utility Pareto fronts. This observation validates our guiding hypothesis that predicting concept labels should not strain the entirety of pretrained LLMs as they are individual classification tasks. Larger LLMs, being repositories of more knowledge through increased parameters, facilitate easier pruning.

- **Limitations of Direct Prompting:** When directly prompting LLMs, such as GPT4 (without fine-tuning on the target datasets), to predict concept and task labels, the resulting performance is noticeably inferior. This highlights the necessity of learning concepts and task labels in target domains. Additionally, since LLMs' task predictions are autoregressively generated and do not rely entirely on the generated concepts, doubts arise regarding the reliability of concept-level explanations.

Explainable Prediction Pathways. The centerpiece of this paper revolves around providing a transparent and interpretive decision-making pathway for each input text vector $\boldsymbol{x} = [t_1, t_2, \cdots, t_d, \cdots, t_D]$, where $t_d, \forall d \in D$ denotes the tokens in the input text. SparseCBMs, at inference time, unravel the following layers of understanding across the decision-making trajectory:

1. **Subnetwork-Level Explanation:** Identification of specific neurons within the LLM backbones responsible for corresponding concepts. This insight is achieved by visualizing individual binary subnetwork masks M_k.

2. **Token-Level Explanation:** Detection of the tokens instrumental in shaping a particular concept. This analysis is carried out by evaluating the gradient of each subnetwork mask with respect to individual tokens $\|\mathcal{G}_k(t_d)\|$.

3. **Concept-Level Explanation:** Understanding the predicted concept labels $\hat{c}_k$ and their contribution to the final prediction. This is captured by computing the dot product between each predicted concept activation and the corresponding weight of the linear predictor: $\phi_k \cdot \hat{\boldsymbol{a}}_k$.

A schematic representation of the decision-making process for a representative example is provided in Figure 2, with "Neg Concept" denoting negative concept values. Additional real-world examples are delineated in Appendix D. Several interesting findings can be drawn from those results:

- **Neural Responsibility Across Concepts:** Various concepts necessitate differing proportions of neurons in the LLM backbone for learning. This resonates with our ambition to demystify the "black-box" LLM backbones by partitioning them into distinct subnetworks, each accountable for an individual concept. Intriguingly, overlaps exist among subnetworks, reflecting that strict disentanglement constraints were not imposed on the backbone parameters. This opens avenues for future research into entirely concept-wise disentangled LLM backbones.

- **Holistic Decision Pathway:** The SparseCBM framework successfully crafts a comprehensive decision-making path that navigates from tokens, through subnetworks and concepts, culminating in the final task label. This rich interpretability paves the way for unique insights into practical applications. For instance, although

concepts like "Food" and "Ambance" may carry identical positive values, the "Food" concept may wield greater influence on the final task label. Additionally, careful examination of parameter masks can shed light on the root causes behind mispredicted concepts, enabling effective and interpretable interventions. We explore this topic further in the subsequent section.

Inference-time Intervention

SparseCBMs distinguish themselves by enabling sparsity-based inference-time intervention, compared to vannila CBMs. This innovative feature creates a pathway for more refined, user-centric interactions by subtly adjusting the masks without the need for direct retraining of the LLM backbone. The significance of this intervention approach lies in its application to real-world scenarios where users often find it easier to articulate broad concepts (e.g., food quality) rather than precise sentiment scores or categorical labels.

Experimental Evaluation. To methodically evaluate this intervention strategy, extensive experiments were conducted on the CEBaB dataset, employing DistilBERT as the representative LLM backbone. The insights gleaned from these experiments apply consistently to other LLMs as well. Figure 3 provides a detailed comparison between concept and task label predictions using SparseCBMs against a baseline, where a vanilla DistilBERT is independently trained to classify concept or task labels. These baseline scores serve as a theoretical upper bound for prediction accuracy, providing a reliable and illustrative benchmark. This analytical exploration not only validates the proposed sparsity-based intervention's efficacy in enhancing inference-time accuracy for both concept and task predictions but also reveals its elegance in execution. With minimal alterations to the underlying model structure, remarkable improvements are achieved. Even for a relatively small model, DistilBERT, the optimal adjustment proportion is found to be a mere 1%, translating to modifications in only 2% of the backbone parameters.

Robustness and Adaptability. These results shed light on the broader applicability and resilience of sparsity-based intervention across various contexts and domains. The capacity to implement such nuanced adjustments without the resource-intensive process of retraining the entire model
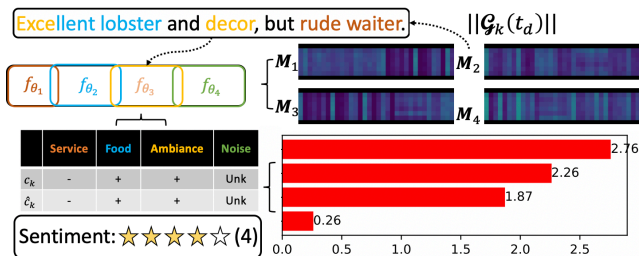

Figure 2: The illustration of a decision pathway of a toy example from the SparseCBM framework with BERT as the backbone. The binary weight masks for each concept is represented as a heatmap.
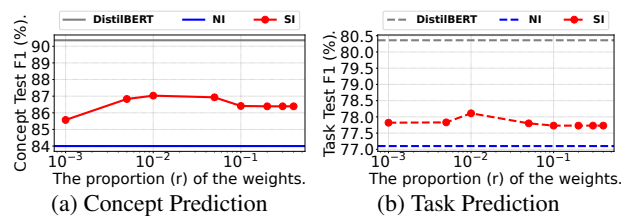

Figure 3: The results of Test-time Intervention. "NI" denotes "no intervention", "SI" denotes "Sparsity-based Intervention". (a) and (b) represent the results for concept and task label prediction respectively. The x-axis indicates the proportion (r) of the weights to perform the intervention.
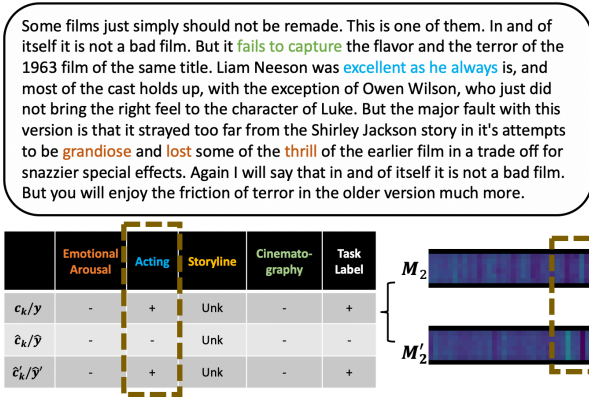
Figure 4: Illustration of the explainable prediction for a real-world example from the `IMDB-C` dataset using OPT-350m as the backbone. The brown boxes with dash lines indicate the test-time intervention on corresponding concepts by modulating the corresponding mask. M_2 and M_2' denote the parameter masks for the second concept, "Acting", before and after the intervention, respectively. We visualize M_2' after seeing all test samples.

marks a substantial advancement toward more agile, responsive machine learning systems. This adaptability resonates with the growing demand for models that can quickly adapt to ever-changing requirements without compromising on performance or interpretability.

Case Study and Insights. To provide an in-depth illustration, a case study depicting the sparsity-based intervention process is presented in Figure 4. This visualization elucidates how the predicted label for the concept "Acting" can be transformed from incorrect " -" to correct "+", subsequently refining the final task label. But the insights run deeper: by visualizing the parameter masks before (M_2) and after (M_2') the intervention, we expose the neural mechanics behind the misprediction and the corrective strategy at the neuron level. This ability to not only correct but also interpret the underlying reasons for prediction errors enhances the overall trustworthiness and usability of the model. In conjunction with the experimental findings, this case study amplifies our understanding of the potential for sparsity-based interventions, not merely as a method for model fine-tuning, but as a principled approach towards more transparent and adaptable AI systems.

Implication. The integration of sparsity-based inference-time intervention within SparseCBMs represents a confluence of accuracy, flexibility, and interpretability. Through careful experimentation and insightful case studies, this work lays the groundwork for models that respond dynamically to the needs of users, augmenting human-machine collaboration in complex decision-making processes. It is a promising step towards building AI models that are not only more effective but also more aligned with the human-centered objectives and ethical considerations of modern machine learning applications.
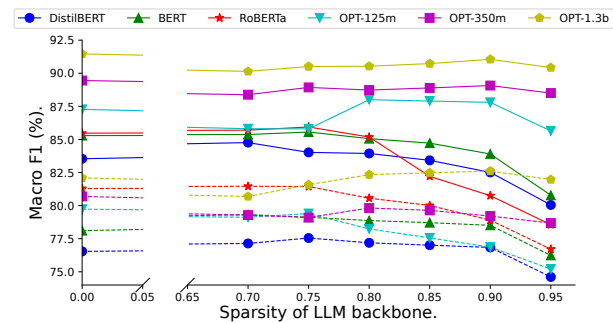

Figure 5: The performance of SparseCBMs across varying LLM backbones in relation to the target sparsity s on the `CEBaB` dataset. Solid lines delineate scores for concept label predictions. Dashed lines capture those for task label predictions. Notably, larger LLM backbones are adept at handling increased sparsity without compromising on prediction efficacy. Nonetheless, excessive pruning invariably impinges on the performance across all LLM backbones.

Sensitivity Analysis on the Sparsity s

In Figure 5, we study the effect of target Large Language Model (LLM) sparsity on concept and task prediction performance across various LLM sizes. The results reveal an interesting trend: larger LLMs tend to have a higher optimal sparsity level compared to smaller ones. This is attributed to the greater knowledge repository and higher redundancy present in larger LLMs, allowing for more extensive pruning without significant performance loss. However, a delicate balance must be struck. While larger LLMs can accommodate more pruning, overdoing it may harm performance. Identifying this balance remains an intriguing avenue for future research, as well as investigating how different pruning strategies interact with various tasks and data distributions.

Conclusion

In this study, we introduced *Sparse Concept Bottleneck Models* (SparseCBMs), a novel method integrating the interpretability of Concept Bottleneck Models (CBMs) with the efficiency of unstructured pruning. By exploiting the properties of second-order pruning, we constructed concept-specific sparse subnetworks in a Large Language Model (LLM) backbone, thereby providing multidimensional interpretability while retaining model performance. Additionally, we proposed a sparsity-driven inference-time intervention mechanism that improves accuracy at inference time, without the need for expensive fine-tuning LLMs. This intervention mechanism effectively identifies the parameters that contribute to each misprediction, enhancing interpretability further. Through rigorous experiments, we demonstrated that SparseCBMs match the performance of full LLMs while offering the added benefits of increased interpretability. Our work underscores the potential of sparsity in LLMs, paving the way for further exploration of this intersection. We envisage future investigations to refine the use of structured sparsity, such as group or block sparsity, to further enhance model transparency and efficiency.

Ethical Statement

This research explores methods to enhance the interpretability and reliability of large language models (LLMs) through the proposed Sparse Concept Bottleneck Models (SparseCBMs). While the development and application of such technology have benefits, including improved model understanding, and more efficient use of computational resources, several considerations arise that warrant discussion.

Transparency and Explainability: Though our work aims to make models more interpretable, the actual understanding of these models can still be quite complex and may be beyond the reach of the general public. Furthermore, the opacity of these models can potentially be exploited, reinforcing the need for ongoing work in model transparency.

Robustness: As indicated in (Tan et al. 2023; Wang et al. 2023b), the proposed framework is sensitive to the noisy concept and target task labels, requesting future work in model robustness. Potential direction include selective learning (Li et al. 2023b,c), knowledge editting (Wang et al. 2023d), to name a few.

Efficiency: It is worthnoteing that, even though the inference-time intervention is highly efficient, SparseCBM require more training time due to the cocnept-specific pruning. Potential way to enhance the training efficiency is to share part of the sparsity among concepts, as studied in (Wang et al. 2020; Chen et al. 2021).

Label Reliance: SparseCBMs, along with other CBM variants, necessitate the annotation of concepts. To reduce this burden, several approaches are promising. These include leveraging other LLMs for automated annotation, as discussed in (Tan et al. 2023; Wang et al. 2023c), employing data-efficient learning techniques (Tan et al. 2022), and exploring the acquisition of implicit concepts through dictionary learning methods (Wang et al. 2022).

Misuse: Advanced AI models like LLMs can be repurposed for harmful uses, including disinformation campaigns or privacy infringement (Jiang et al. 2023; Chen and Shu 2023). It's crucial to implement strong ethical guidelines and security measures to prevent misuse.

Automation and Employment: The advancements in AI and machine learning could lead to increased automation and potential job displacement. We must consider the societal implications of this technology and work towards strategies to manage potential employment shifts.

Data Bias: If the training data contains biases, LLMs may amplify these biases and result in unfair outcomes. We need to continue to develop methods to mitigate these biases in AI systems and promote fair and equitable AI use.

In conducting this research, we adhered to OpenAI's use case policy and are committed to furthering responsible and ethical AI development. As AI technology advances, continuous dialogue on these topics will be needed to manage the potential impacts and ensure the technology is used for the betterment of all.

Acknowledgments

References

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. *NeurIPS*, 35: 17582–17596.

Chen, C.; and Shu, K. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv:2311.05656*.

Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; and Wang, Z. 2021. Long live the lottery: The existence of winning tickets in lifelong learning. In *ICLR*.

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

Evci, U.; Gale, T.; Menick, J.; Castro, P. S.; and Elsen, E. 2020. Rigging the lottery: Making all tickets winners. In *ICML*, 2943–2952.

Galassi, A.; Lippi, M.; and Torroni, P. 2020. Attention in natural language processing. *IEEE transactions on neural networks and learning systems*, 32(10): 4291–4308.

Gale, T.; Elsen, E.; and Hooker, S. 2019. The state of sparsity in deep neural networks. *arXiv:1902.09574*.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both Weights and Connections for Efficient Neural Network. In *NeurIPS*, 1135–1143.

Hassibi, B.; and Stork, D. 1992. Second order derivatives for network pruning: Optimal brain surgeon. *NeurIPS*, 5.

He, Y.; Zhang, X.; and Sun, J. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of ICCV*.

Jiang, B.; Tan, Z.; Nirmal, A.; and Liu, H. 2023. Disinformation Detection: An Evolving Challenge in the Age of LLMs. *arXiv:2309.15847*.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept bottleneck models. In *ICML*, 5338–5348.

Kurtic, E.; Campos, D.; Nguyen, T.; Frantar, E.; Kurtz, M.; Fineran, B.; Goin, M.; and Alistarh, D. 2022. The Optimal BERT Surgeon: Scalable and Accurate Second-Order Pruning for Large Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 4163–4181.

Lagunas, F.; Charlaix, E.; Sanh, V.; and Rush, A. M. 2021. Block pruning for faster transformers. *arXiv:2109.04838*.

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1990a. Optimal brain damage. In *NeurIPS*, 598–605.

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1990b. Optimal Brain Damage. In Touretzky, D. S., ed., *NeurIPS*, 598–605. Morgan-Kaufmann.

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

Li, S.; Chen, J.; Shen, Y.; Chen, Z.; Zhang, X.; Li, Z.; Wang, H.; Qian, J.; Peng, B.; Mao, Y.; et al. 2022. Explanations from large language models make small reasoners better. *arXiv:2210.06726*.

Li, Y.; Han, H.; Shan, S.; and Chen, X. 2023b. DISC: Learning from Noisy Labels via Dynamic Instance-Specific Selection and Correction. In *Proceedings of the IEEE/CVF Conference on CVPR*, 24070–24079.

Li, Y.; Tan, Z.; Shu, K.; Cao, Z.; Kong, Y.; and Liu, H. 2023c. CSGNN: Conquering Noisy Node labels via Dynamic Class-wise Selection. *arXiv:2311.11473*.

Liu, J.; Lin, Y.; Jiang, L.; Liu, J.; Wen, Z.; and Peng, X. 2022. Improve Interpretability of Neural Networks via Sparse Contrastive Coding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 460–470.

Liu, S.; Chen, T.; Zhang, Z.; Chen, X.; Huang, T.; Jaiswal, A.; and Wang, Z. 2023. Sparsity May Cry: Let Us Fail (Current) Sparse Neural Networks Together! *arXiv:2303.02141*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of ICCV*, 2736–2744.

Losch, M.; Fritz, M.; and Schiele, B. 2019. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv:1907.10882*.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *NeurIPS*, 30.

Meister, C.; Lazov, S.; Augenstein, I.; and Cotterell, R. 2021. Is Sparse Attention more Interpretable? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 122–129.

Michel, P.; Levy, O.; and Neubig, G. 2019. Are sixteen heads really better than one? *NeurIPS*, 32.

Mishra, S.; Sturm, B. L.; and Dixon, S. 2017. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, volume 53, 537–543.

Oikarinen, T.; Das, S.; Nguyen, L. M.; and Weng, T.-W. 2023. Label-free Concept Bottleneck Models. In *ICLR*.

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

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. ” Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD Conference*, 1135–1144.

Ross, A.; Marasović, A.; and Peters, M. E. 2021. Explaining NLP Models via Minimal Contrastive Editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3840–3852.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*.

Subramanian, A.; Pruthi, D.; Jhamtani, H.; Berg-Kirkpatrick, T.; and Hovy, E. 2018. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, Z. 2023. A Simple and Effective Pruning Approach for Large Language Models. *arXiv:2306.11695*.

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

Tan, Z.; Ding, K.; Guo, R.; and Liu, H. 2022. Graph few-shot class-incremental learning. In *Proceedings of the Fifteenth ACM International Conference on WSDM*, 987–996.

Vig, J. 2019. A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 37–42.

Wang, H.; Hong, Z.; Zhang, D.; and Wang, H. 2023a. Intepreting & Improving Pretrained Language Models: A Probabilistic Conceptual Approach. Openreview:id=kwF1ZfHf0W.

Wang, P.; Fan, Z.; Chen, T.; and Wang, Z. 2022. Neural implicit dictionary learning via mixture-of-expert training. In *ICML*, 22613–22624.

Wang, S.; Tan, Z.; Guo, R.; and Li, J. 2023b. Noise-Robust Fine-Tuning of Pretrained Language Models via External Guidance. *arXiv:2311.01108*.

Wang, S.; Tan, Z.; Liu, H.; and Li, J. 2023c. Contrastive Meta-Learning for Few-shot Node Classification. In *Proceedings of the 29th ACM SIGKDD Conference*, 2386–2397.

Wang, S.; Zhu, Y.; Liu, H.; Zheng, Z.; Chen, C.; et al. 2023d. Knowledge Editing for Large Language Models: A Survey. *arXiv:2310.16218*.

Wang, Z.; Jian, T.; Chowdhury, K.; Wang, Y.; Dy, J.; and Ioannidis, S. 2020. Learn-prune-share for lifelong learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, 641–650. IEEE.

Wu, T.; Ribeiro, M. T.; Heer, J.; and Weld, D. 2021. Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. In *Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*.

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:2205.01068*.

Zhou, H.; Alvarez, J. M.; and Porikli, F. 2016. Less is more: Towards compact cnns. In *ECCV*, 662–677. Springer.

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 *ICLR*.