OVERTHINKING THE TRUTH: UNDERSTANDING HOW LANGUAGE MODELS PROCESS FALSE DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Through few-shot learning or chain-of-thought prompting, modern language models can detect and imitate complex patterns in their prompt. This behavior allows language models to complete challenging tasks without fine-tuning, but can be at odds with completion quality: if the context is inaccurate or harmful, then the model may reproduce these defects in its completions. In this work, we show that this harmful context-following appears late in a model's computation–in particular, given an inaccurate context, models perform *better* after zeroing out later layers. More concretely, at early layers models have similar performance given either accurate and inaccurate few-shot prompts, but a gap appears at later layers (e.g. layers 10-14 for GPT-J). This gap appears at a consistent depth across datasets, and coincides with the appearance of "induction heads" that attend to previous answers in the prompt. We restore the performance for inaccurate contexts by ablating a subset of these heads, reducing the gap by 28% on average across 8 datasets. Our results suggest that studying early stages of computation could be a promising strategy to prevent misleading outputs, and that understanding and editing internal mechanisms can help correct unwanted model behavior.

1 INTRODUCTION

A key behavior of modern language models is context-following: neural networks like GPT-3 are able to infer and imitate the patterns in their prompt. At its best, this allows language models to perform well on benchmarks without the need for fine-tuning (Brown et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Chowdhery et al., 2022; Srivastava et al., 2022). This has led researchers to study how properties of the context affect few-shot performance (Min et al., 2022b; Kim et al., 2022; Xie et al., 2021; Zhao et al., 2021), and what internal mechanisms underlie context-following (Olsson et al., 2022).

However, context-following can also lead to incorrect, toxic or unsafe model outputs (Rong, 2021). For example, if an inexperienced programmer prompts Codex (Chen et al., 2021) with poorly written or vulnerable code, the model is likely to produce poorly written or vulnerable code completions. Similarly, in this work we study few-shot learning for classification tasks: prompting the model with inaccurate demonstrations reduces model accuracy (Figure 1, left), because the model learns to reproduce the false demonstrations. We thus ask: Can we attribute this "false context-following" behavior to specific model components, and can we mitigate it by intervening on these components?

We show that, perhaps surprisingly, false context-following in text classification is primarily a property of late stages of computation. In particular, stopping the model early–by zeroing out the later layers (Nostalgebraist, 2020)–actually *improves* performance (Figure 1, center). Moreover, true and false contexts yield similar accuracy until some "critical layer" at which they sharply diverge. This demonstrates that even with false demonstrations, the model often "knows" the correct answer (it can be easily decoded from the latent states) but later replaces it with an incorrect answer that is more likely given the context.

To identify the underlying mechanism for false context-following, we turn to Olsson et al. (2022), who identify "induction heads" that attend to and reproduce previous patterns in the input. Motivated by this, we searched for heads that consistently attend to previous examples that have the same (true)

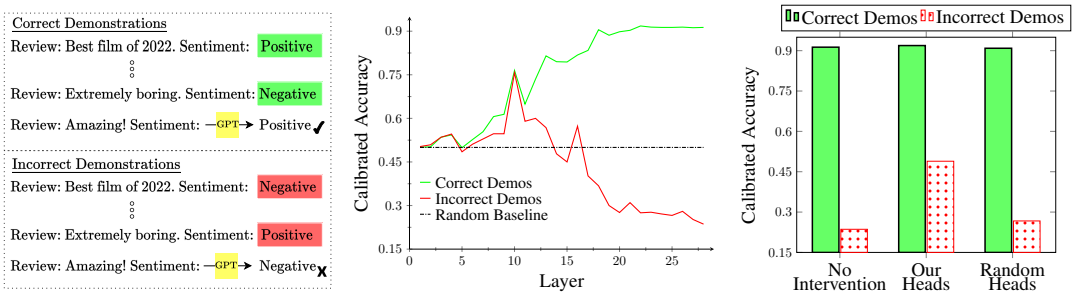


Figure 1: **Left:** Given a prompt of inaccurate demonstrations, language models are more likely to output incorrect labels. **Center:** When demonstrations are incorrect, zeroing out the later layers increases the classification accuracy, here on SST-2. **Right:** We identify 15 attention heads and remove them from the model: this reduces the effect of incorrect demonstrations by 36% on SST-2, without decreasing the accuracy given correct demonstrations.

answer as the current prompt. We found many such heads, primarily concentrated in later layers of the model (after the critical layer). By removing 15 of these heads, we are able to reduce the accuracy gap between accurate and inaccurate prompts by an average of 28% over 8 datasets, with negligible effects on the performance given true prefixes (Figure 1, right).

Our findings show how analyzing and editing model internals can help practictioners understand and mitigate model failures. Indeed, one intuition for why early-exiting succeeds is that the attention heads we identified cannot in general occur at the earliest layers. This is because these heads must recognize which inputs belong to the same class, which likely requires multiple layers of processing. Thus, early exiting might be a generally promising strategy to detect dishonest behavior in models.

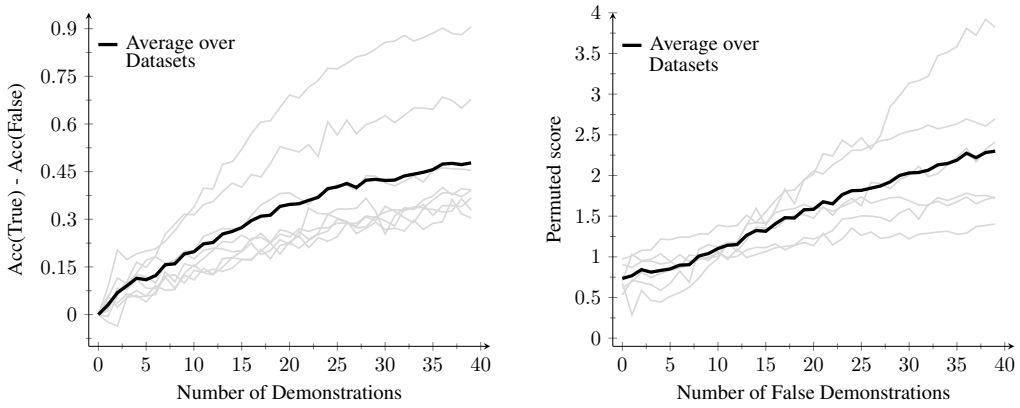
2 Preliminaries: Few-shot Learning with False Demonstrations

We begin by introducing the setting we study: few-shot learning for classification, given demonstrations with correct or incorrect labels. Incorrect demonstrations consistently reduce classification performance, which is the phenomenon that we aim to study and mitigate in this work.

Few-shot learning. We consider autoregressive transformer language models, which produce a conditional probability distribution $p(t_{n+1} \mid t_1, ..., t_n)$ over the next token t_{n+1} given previous tokens. We focus on the few-shot learning setting (Brown et al., 2020) for classification tasks: we sample k demonstrations (input-label pairs) from the task dataset, denoted $(x_1, y_1), ..., (x_k, y_k)$. To query the model on a new input x, we use the predictive distribution $p(y \mid x_1, y_1, ..., x_k, y_k, x)$.

Datasets and models. We consider seven text classification datasets: SST-2 (Socher et al., 2013), AGNews (Zhang et al., 2015), TREC (Voorhees & Tice, 2000), DBpedia (Zhang et al., 2015), TweetEval-Hate (Barbieri et al., 2020), SICK (Marelli et al., 2014), and Poem Sentiment (Sheng & Uthus, 2020). We used the same prompt formats as in Min et al. (2022b) and Zhao et al. (2021) (for SST-2 we use the first of the 15 prompt formats in Zhao et al.). We evaluated 3 autoregressive language models: GPT-J (Wang & Komatsuzaki, 2021), GPT2-XL (Radford et al., 2019), and GPT-NeoX-20B (Black et al., 2022).

Evaluation metrics. Given our focus on classification tasks, we are interested in how often the model assigns higher probability to the true label than to all other labels. However, model predictions can be very unstable with respect to small prompt perturbations (Gao et al., 2021). To mitigate this variability, we measure the *calibrated* classification accuracy (Zhao et al., 2021). Concretely, for a 2-class classification task, we measure how often the correct label has a higher probability than its median probability over the dataset. Assuming the dataset is balanced (which is true for us), this step has been shown to improve performance and reduce variability across prompts. Calibration for multi-class tasks follows a similar procedure, detailed in appendix A.2.



(a) (b)

Figure 2: **(a)** The difference in accuracy between accurate and inaccurate prompts increases with the number of demonstrations. **(b)** As the number of false demonstrations increases, the model chooses the permuted label $\sigma(\text{class}(x))$ more often than the other labels, rather than making random errors

2.1 Comparing true and false demonstrations

We first confirm that the models we study exhibit false context-following behavior. To do so, we compare the performance of models when the demonstration labels are all correct, i.e. $y_i = \text{class}(x_i)$, and when they are all incorrect, i.e. $y_i = \sigma(\text{class}(x_i))$, for a cyclic permutation σ over the set of labels (Figure 1, left). In particular, inputs from the same class are always assigned the same (possibly false) label within each prompt.

For each model and dataset, we sample 100 sequences each containing k demonstrations and evaluate the model's calibrated accuracy. We sample different demonstrations (x_i, y_i) and label permutations σ for every sequence, and vary k from 0 to 40 (from 0 to 25 for GPT2-XL, due to its smaller context size). We repeat this 10 times, and average the calibrated accuracies over the 10 repetitions.

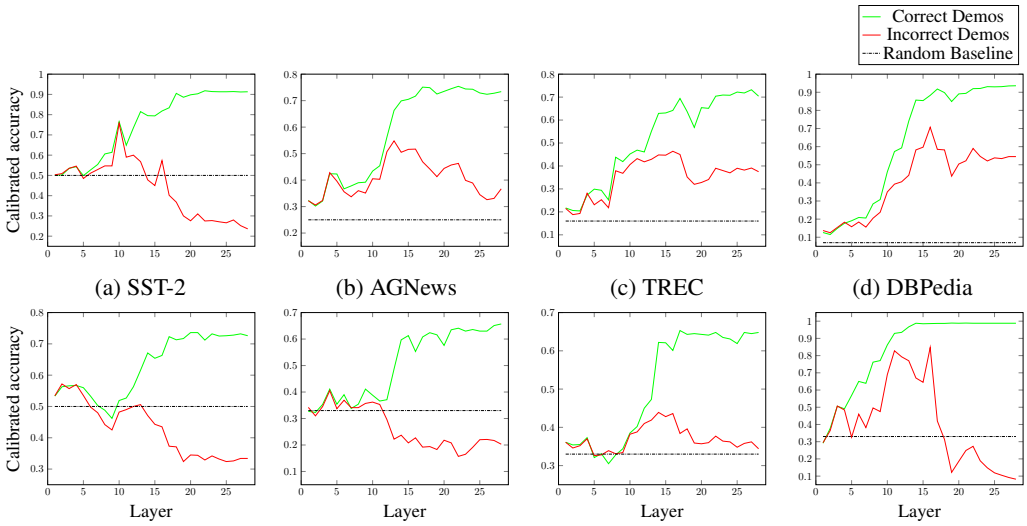
Figure 2a shows the difference between GPT-J's calibrated accuracy given accurate and inaccurate prompts as the number of demonstrations increases. As expected, false demonstrations lead to worse performance, and the gap tends to increase with k for most datasets. These results are in agreement with Min et al. (2022b), who found that incorrect demonstrations decreased GPT-J's performance on classification tasks (see Figure 4 in Min et al.).

Models could lose accuracy by copying the incorrect label, or by becoming confused and choosing random labels. To confirm it is the former, we also measure which labels the model chooses for multi-class tasks. Specifically, we measure the *permuted score*: how often the model chooses the permuted label $\sigma(\text{class}(x))$ over the other labels. For each dataset, a random classifier would have a permuted score of $\frac{1}{\#\text{labels}}$. To make the results comparable across datasets, we divide the permuted scores by this random baseline. Figure 2b shows these reweighted permuted scores for GPT-J on the 6 multi-class datasets in our collection, as well as their average over the datasets. The permuted score increases steadily with the number of demonstrations and reaches twice its baseline value after 40 demonstrations.

3 The Logit Lens: Zeroing Out Later Layers Improves Accuracy

In this section, we decode model predictions directly from intermediate layers. This allows us to evaluate the model's performance midway through processing the inputs. On false prefixes, we find that the model performs *better* midway through processing, and investigate this phenomenon in detail.

Intermediate layer predictions: the logit lens. Given an autoregressive transformer language model, we will decode a probability distribution over the next token from each intermediate layer,



(a) SST-2 (b) AGNews (c) TREC (d) DBPedia

(e) TE-Hate (f) SICK (g) PoemSentiment (h) Unnatural

Figure 3: GPT-J early-exit classification accuracies across 8 datasets, given correct and incorrect demonstrations. Given incorrect demonstrations, zeroing out all transformer blocks after layer 16 outperforms running the entire model, across all 8 datasets.

using the "logit lens" method (Nostalgebraist, 2020). Intuitively, these intermediate distributions represent model predictions after $\ell \in \{1, ..., L\}$ layers of processing.

In more detail, let $h_\ell^{(i)} \in \mathbb{R}^d$ denote the hidden state of token t_i at layer ℓ, i.e. the sum of everything up to layer ℓ in the residual stream. For a sequence of tokens $t_1, ..., t_n \in V$, the logits of the predictive distribution $p(t_{n+1} \mid t_1, ..., t_n)$ are given by

$$[\text{logit}_1, ..., \text{logit}_{|V|}] = W_U \cdot \text{LayerNorm}_L(h_L^{(n)}),$$

where LayerNorm_L is the the pre-unembedding layer normalization, and $W_U \in \mathbb{R}^{|V| \times d}$ is the unembedding matrix. The logit lens applies the same unembedding operation to the earlier hidden states $h_\ell^{(i)}$, yielding an intermediate layer distribution $p_\ell(t_{n+1} \mid t_1, ..., t_n)$:

$$[\text{logit}_1^\ell, ..., \text{logit}_{|V|}^\ell] = W_U \cdot \text{LayerNorm}_L(h_\ell^{(n)}).$$

This provides a measurement of what predictions the model represents at layer ℓ, without the need to train a new decoding matrix. It can therefore be interpreted as a form of early exiting (Panda et al., 2015; Teerapittayanon et al., 2017; Figurnov et al., 2017).

Early exiting improves classification performance. We measure the calibrated accuracies of the intermediate layer distributions p_ℓ for the three models and seven datasets from Section 2, using context lengths of 40 demonstrations (25 demonstrations for GPT2-XL). We also measure the layerwise accuracies for a toy dataset, "Unnatural", that extends a task in Rong (2021; section 4). In this dataset, demonstrations are of the form "[object]: [label]" and the labels are "plant/vegetable", "sport", and "animal".

Figure 3 displays results for GPT-J, with corresponding plots for GPT2-XL and GPT-NeoX in Figures 8 and 9 in the Appendix. For GPT-J with correct demonstrations, accuracy tends to increase with layer depth, and starts to stagnate or grow more slowly around layer 15. The accuracy for incorrect demonstrations follows a similar trend at the early layers, but then diverges and decreases at the later layers.

For incorrect demonstrations, decoding from earlier layers performs *better* than decoding from the final layer. For GPT-J, using p_{16} (the first 16 layers) achieves a better accuracy than the full model for all 8 datasets, by an average of 18 percentage points. Similarly, for GPT2-XL and GPT-NeoX, the intermediate predictions p_{30} and p_{23} respectively outperform using the full model, for 7 out of

Figure 4: Examples of attention patterns on incorrect demonstrations from the "unnatural" dataset, for heads with high label-attending but low class-sensitivity scores (Left), low label-attending but high class-sensitivity scores (Center), and high label-attending and class-sensitivity scores (Right).

8 and 6 out of 8 datasets respectively. This gain on false prefixes comes with a comparatively small cost for true prefixes: 4 percentage points on average.

For the toy "Unnatural" dataset, these effects are particularly pronounced (see Figure 3h). At layer 16, the accuracy of GPT-J for incorrect demonstrations reaches 0.85, which is 86% of the final layer accuracy given an accurate prompt. In contrast, at the final layer, GPT-J's accuracy given false demonstrations reaches its lowest value, 0.08.

True and false prefixes sharply diverge at "critical layers". For each model, the accuracies for correct and incorrect demonstrations diverge at the same layers across all datasets. For example, for GPT-J, the accuracy gap between accurate and inaccurate prompts reaches 35% of its final layer value at layers 13 or 14 for all 8 datasets. The same is true for GPT-NeoX with layers 9 and 10, and for GPT2-XL with layers 20 to 21. In summary, zeroing out later layers leads to better classification accuracies given incorrect demonstrations, and the accuracy gap between correct and incorrect demonstrations emerges at a consistent set of layers across datasets.

4 Zooming into attention heads

We found that for all datasets, the gap between true and false demonstrations appears in a small set of transformer blocks. We would like to know whether some specific attention heads are responsible for this behavior.

(Olsson et al., 2022) introduce *induction heads*: attention heads that attend to previous occurences of the present token, and increase the probability of the outputs that follow them. Inspired by this work, we investigate the hypothesis that a small number of "induction heads" play a key role in false context-following, by attending to the labels in previous similar demonstrations and making the model more likely to output them.

For example, in Figure 4, we know that the model assigns a high probability to the mistaken label "sport". According to the hypothesis, this is because of heads that attend to the previous occurences of "sport" in this context, and increase the probability of that token. The previous occurences of "sport" share two properties: (1) they are *labels* in the previous demonstrations, and (2) they follow inputs *with the same class* as "beet": "tomato" and "garlic".

Therefore, we look for heads that satisfy two conditions when they attend to inaccurate prompts. First, they should be *label-attending*, i.e. concentrate their attention on labels in the previous demonstrations. Second, they should be *class-sensitive*, meaning they should attend specifically to those labels that follow inputs in the same class as the latest input. We call heads that are both label-attending and class-sensitive given incorrect demonstrations *false prefix-matching heads*.

We define two scores to quantify how label-attending and class-sensitive a head is. For a sequence of demonstrations (x_i, y_i) and a final input x,

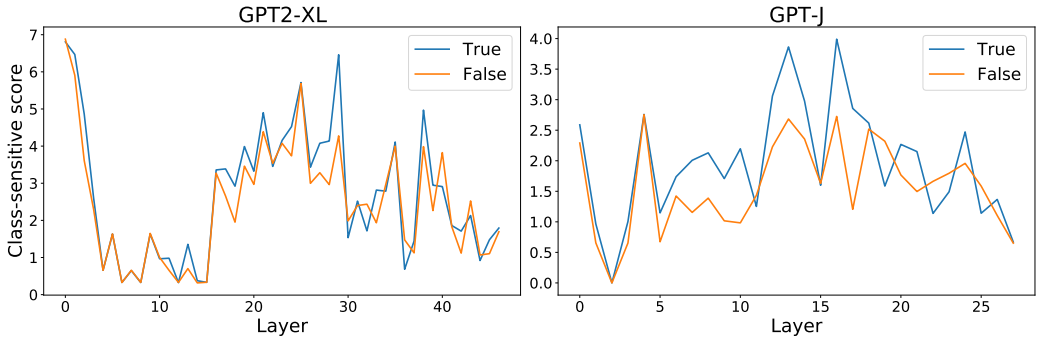


Figure 5: Sum of class-sensitivity scores for heads with the top 25% label-attending scores, for true and false demonstrations, for GPT2-XL (left) and GPT-J (right). The CSS score increases around the layers where the accuracy gap between true and false demonstrations emerges.

- the **label-attending score** (LAS^h) of a head h is the sum of attention weights between the final token of x and the tokens corresponding to the labels y_i:

$$\text{LAS}^h = \sum_{i=1}^{n} \text{Att}^h(x, y_i).$$

- the **class-sensitivity score** (CSS^h) of a head is the fraction of attention weights between x and the y_i where x_i belongs to the same class as x:

$$\text{CSS}^h = \frac{1}{\text{LAS}^h} \sum_{i=1}^{n} \text{Att}^h(x, y_i) \cdot \mathbf{1}\{\text{class}(x) = \text{class}(x_i)\}.$$

Prefix-matching heads should have both high LAS and CSS scores. We therefore: (1) restrict to the 25% of heads with the highest label-attending score (averaged across some dataset), and (2) plot the distribution of class-sensitivity scores across layers for these heads. Figure 5 shows results for the "Unnatural" dataset, for both true and false demonstrations. For each model, the scores remain low at the early layers (apart from the first 2 layers in GPT2-XL), but then increase around the "critical layers" that we identified in the previous section. This lends correlational support to our hypothesis that false prefix-matching heads cause false-context following behavior.

Ablating false prefix-matching heads. However, we are interested in causal evidence. Therefore, we check whether removing false prefix-matching heads reduces false context-following. We select 15 heads from GPT-J with high label-attending and class-sensitivity scores for the false prefix on the "Unnatural" dataset, as detailed in Appendix A.1. We ablate these heads by setting their keys, queries and values to zero. We then evaluate this lesioned model on all 8 datasets, and compare its layerwise performance to the original model's. As a control baseline, we also perform the same analysis for 15 heads selected at random.

The ablations considerably increase the accuracy given false demonstrations: they reduce the gap in accuracy between accurate and inaccurate prompts by an average of 28.3% for $k = 40$ and 37.5% for $k = 10$ (see Table 1). In contrast, ablating random heads reduces the gap by 2.97% for $k = 40$ and -0.24% for $k = 10$. While they greatly improve the accuracy given a false prefix, our ablations have a comparatively small effect on the accuracy given correct demonstrations: ablating the false prefix-matching heads decreases the accuracy given true demonstrations by 2% for $k = 40$ and by 0.31% for $k = 10$. These results show that the false prefix-matching heads cause a large fraction of the false context-following behavior.

Analysing the outputs of false prefix-matching heads. We identified false prefix-matching heads based only on their attention patterns. However, our postulated mechanism also depends on the heads' outputs: they must increase the probability of the labels they attend to. We therefore study the outputs of these heads to understand how they affect the residual stream.

We apply the logit lens to each head individually, by applying layer normalization followed by the unembedding matrix to its outputs. This tells us how much the head increases or decreases the

Table 1: Ablating false prefix-matching heads recovers a large fraction of the accuracy gap between true and false prefixes, without hurting performance given true prefixes. We show the percentage reduction of the accuracy gap and percentage change in true prefix performance when ablating the 15 false prefix-matching heads chosen using the Unnatural dataset ("top") or 15 random heads ("random"). We bold gap reductions when they are greater for our heads than for the random heads.

Dataset	**Heads**	**10-shot**		**40-shot**	
		True Prefix %Δ(↑)	Gap Reduction %(↑)	True Prefix %Δ(↑)	Gap Reduction %(↑)
Sentiment Analysis					
SST-2	top	0.72	**35.9**	0.76	**37.3**
	random	−3.86	2.86	−0.43	4.57
Poem Sent.	top	3.72	**44.2**	−0.46	**36.8**
	random	−1.48	1.28	−1.23	−1.64
Hate Speech Detection					
Tweet Hate	top	1.32	**20.5**	−5.37	**16.8**
	random	1.49	10.5	−3.30	2.04
Natural Language Inference					
SICK	top	−14.2	3.50	−10.3	**2.64**
	random	−3.61	3.50	−4.87	1.98
Topic Classification					
AGNews	top	2.81	**35.2**	−2.70	**40.0**
	random	0.49	−7.19	−0.68	3.54
Trec	top	3.15	**57.7**	2.27	**17.9**
	random	2.65	11.0	−0.42	4.86
DBPedia	top	−0.22	**29.1**	−0.10	**29.1**
	random	−1.00	−7.42	−0.42	3.58
Unnatural	top	0.20	**73.0**	−0.10	**46.0**
	random	−0.50	−16.5	−0.30	4.85
Average	top	−0.31	**37.5**	−2.00	**28.3**
	random	−0.73	−0.24	−1.46	2.97

intermediate logits of each token. We then measure each head's *permuted score* (see 2): how often it increases the logits of the permuted label $\sigma(\text{class}(x))$ on the Unnatural dataset more than the other labels. Our 15 false prefix-matching heads have an average permuted score of 0.46, which is greater than the baseline value of $\frac{1}{\#\text{label}} = 1/3$. Moreover, when sampling 1000 sets of 15 random heads, we find an average permuted score of 0.37, with a standard deviation of 0.03. Therefore, false prefix-matching heads directly increase the probability of the permuted labels more often than random heads. However, this behavior is far from consistent across inputs, which suggests that token copying does not explain the totality of the accuracy gap between true and false demonstrations.

5 Discussion and Related Work

In this paper, we showed how stopping language models early by zeroing out their later layers improves classification performance given inaccurate contexts, without requiring any additional training. We also identified attention heads that contribute to the effect of misleading prompts, and showed that ablating these heads mitigates this effect.

Related work. Our work is closely related to Min et al. (2022b) and Kim et al. (2022), who examine the role of false demonstration on model accuracy. Min et al. (2022b, figure, 4) find that for classification by a pre-trained model (GPT-J), the ground truth of demonstrations has a large

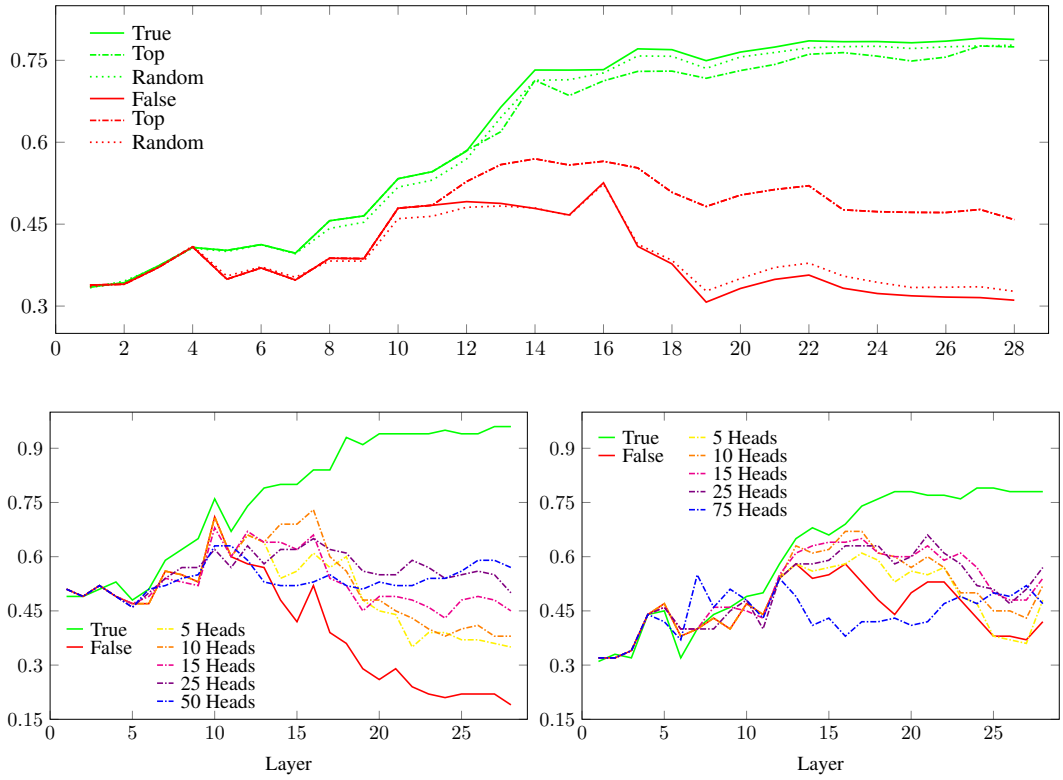


Figure 6: Ablating false prefix-matching heads increases accuracy across multiple layers. **Top:** Average accuracy at each layer before and after ablating false-prefix matching or random heads, given correct and incorrect demonstrations. **Bottom:** Accuracy at each layer for incorrect demonstrations on SST-2 and AGNews, after ablating the k most class-sensitive heads, for $k \in \{5, 10, 15, 35, 75\}$.

effect on the accuracy. However, they do not find such an effect with a meta-tuned model (Min et al., 2022a). Therefore, meta-tuning could serve as "negative control" to test our hypothesis that false prefix-matching heads cause false context-following: an interesting future direction would be to check whether meta-tuning reduces the number of false prefix-matching heads.

The literature on early-exiting and overthinking (Kaya et al., 2018; Panda et al., 2015; Teerapittayanon et al., 2017; Figurnov et al., 2017; Hou et al., 2020; Liu et al., 2020; Xin et al., 2020; Zhou et al., 2020; Zhu, 2021; Schuster et al., 2022) also highlights how decoding from intermediate layers can save compute and sometimes produce better results. One major difference is that most of these methods rely either on modifying the training process to allow for early-exit, or on training additional probes to decode intermediate states. In contrast, the logit lens does not require any extra training to decode answers from internal representations.

How does the logit lens compare to probing? Our work, especially Section 3, relies heavily on the "logit lens" (Nostalgebraist, 2020). We find it useful to think of this method in comparison to probing.

If a layer has a high probing accuracy, this means that the correct answer can be decoded from the hidden states. However, this is often a low bar to clear, especially when the classification task is easy and the hidden states are high-dimensional (Hewitt & Liang, 2019). In contrast, if a layer has a high logit lens accuracy, this shows that it encodes correct answers along a direction in the residual stream that the model subsequently decodes from, which is much more informative. On the other hand, a low logit lens performance at a layer does not imply that the correct answers cannot be decoded from that layer.

One intermediate between probing and zeroing out later layers is the "tuned lens" (Ostrovsky et al., 2022): instead of training probes on each classification task or directly using the final layer's decoding matrix, for each layer the authors train a new square adapter matrix between the residual stream and the unembedding matrix on a language modelling dataset such as the Pile (Gao et al., 2020). It would be interesting to run our experiments with this alternative decoding method.

Future work. While we find consistent results across 8 datasets, our experiments are restricted to a specific setting: text classification with a large number of incorrect few-shot examples. In the future, researchers could use the logit lens to study diverse real-world failures of large models, such as "prompt injection" (Branch et al., 2022) or vulnerable code completions (Pearce et al., 2022). In both of these cases, the model outputs inaccurate or harmful completions even though it is capable of producing correct ones given a better prompt.

In addition, our head ablations do not recover the entirety of the accuracy gap between accurate and inaccurate prompts. This could be because we did not identify some of the model components that cause false context-following. However, there is another possibility: if an attention head's outputs are on average far from zero, zeroing out that head takes the intermediate states off-distribution, which can decrease overall model performance. Thus, one promising future direction would be to replace head outputs by their value on different inputs, as in (Meng et al., 2022).

Relatedly, while we identified induction heads that increase the probability of the wrong answers, we still do not have a full mechanistic understanding of false context-following behavior. For example, we do not know which heads in the earlier layers compose to form these induction heads. Future work could build on our methodology to reverse-engineer circuits (Cammarata et al., 2020) in GPT models that implement false context-following.

In this work we showed how studying the early stages of computation can mitigate the effects of misleading prompts. We hope this will spur further work on auditing network internals to detect dishonest behavior in models.

REFERENCES

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, nov 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL https://aclanthology.org/2020.findings-emnlp.148.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022. URL https://arxiv.org/abs/2204.06745.

Hezekiah J. Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias, Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples, 2022. URL https://arxiv.org/abs/2209.02128.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. https://distill.pub/2020/circuits.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Michael Figurnov, Artem Sobolev, and Dmitry Vetrov. Probabilistic adaptive computation time, 2017.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. 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)*, pp. 3816–3830, Online, aug 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295.

J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. URL https://nlp.stanford.edu/pubs/hewitt2019control.pdf.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth, 2020.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking, 2018.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang goo Lee, Kang Min Yoo, and Taeuk Kim. Ground-truth labels matter: A deeper look into input-label demonstrations, 2022.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and QI JU. Fastbert: a self-distilling bert with adaptive inference time. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.537. URL http://dx.doi.org/10.18653/v1/2020.acl-main.537.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022a. doi: 10.18653/v1/2022.naacl-main.201. URL http://dx.doi.org/10.18653/v1/2022.naacl-main.201.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022b.

Nostalgebraist. Interpreting gpt: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Igor Ostrovsky, Ram Bharadwaj Aryasomayajula, Dakota Mahan, and Stella Biderman. Towards 20/20 vision: Interpreting transformers with tuned lenses, forthcoming (private communication), 2022.

Priyadarshini Panda, Abhronil Sengupta, and Kaushik Roy. Conditional deep learning for energy-efficient and enhanced pattern recognition, 2015.

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. Asleep at the keyboard? assessing the security of github copilot's code contributions. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 754–768. IEEE, 2022.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Frieda Rong. Extrapolating to unnatural language processing with gpt-3's in-context learning: The good, the bad, and the mysterious, 2021. URL https://ai.stanford.edu/blog/in-context-learning/.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling, 2022.

Emily Sheng and David Uthus. Investigating societal biases in a poetry composition system, 2020. URL https://arxiv.org/abs/2011.02686.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks, 2017.

Ellen M. Voorhees and Dawn M. Tice. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May 2000. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2000/pdf/26.pdf.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference, 2021.

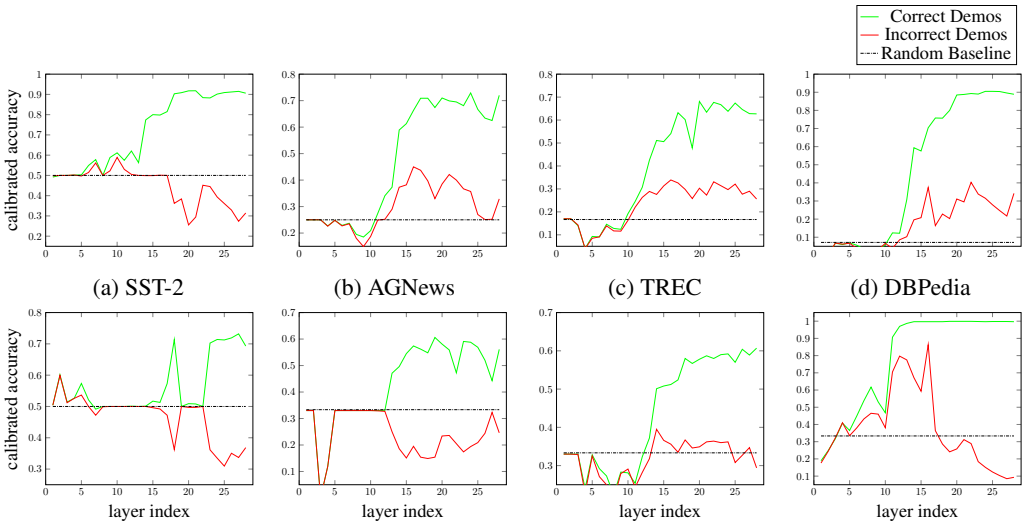
Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference, 2020.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2015. URL https://arxiv.org/abs/1509.01626.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models, 2021.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit, 2020.

Wei Zhu. Leebert: Learned early exit for bert with cross-level optimization. In *ACL*, 2021.



(a) SST-2 (b) AGNews (c) TREC (d) DBPedia

(e) TE-Hate (f) SICK (g) PoemSentiment (h) Unnatural

Figure 7: GPT-J early-exit *uncalibrated* classification accuracies across 8 datasets, given correct and incorrect demonstrations. The lack of calibration makes the results noisier especially at early layers, but early-exit still generally outperforms running the full model.

A Appendix

A.1 Head choice

In this section, we describe how we choose the 15 heads we ablate in our experiments. In summary, we select heads to have large label-attending and class-sensitivity scores, with a bias towards heads in the later layers. To choose the first 10 heads, we consider the 25 heads with the highest label-attending score, and the select the 10 heads with the highest class-sensitivity scores among these.

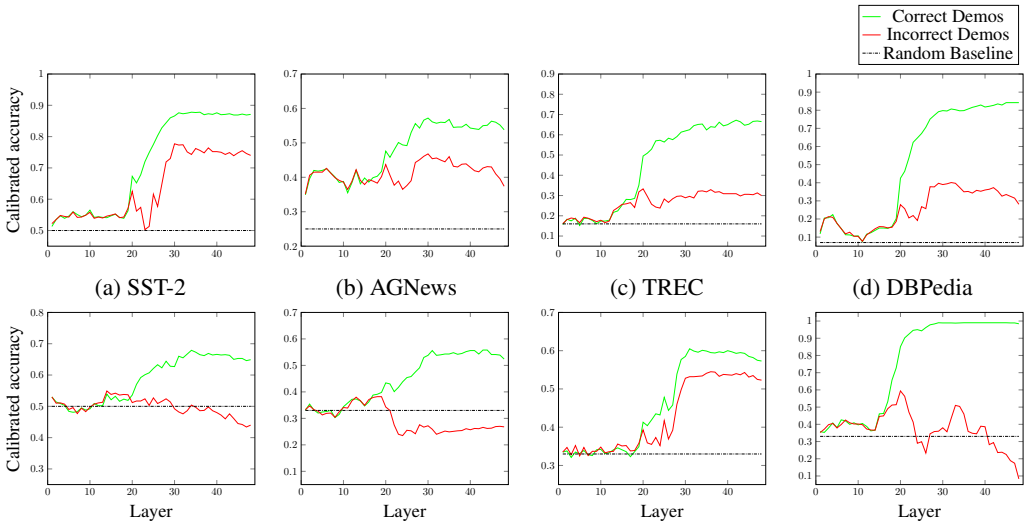
These 10 heads belong to layers 20 and less, yet the logit lens results often show a sharp permuted score increase around layer 20. Thus, to add the other 5 heads, we consider the 50 heads with the highest label-attending scores, and the 25 among these with the highest class-sensitivity scores. We then select the 5 heads in this set of 25 that belong to layers 20 and later.

A.2 Calibration

For k-way tasks, we measure how often the correct label has a higher probabilitiy than the $\frac{k-1}{k}$-quantile of its probability over the dataset. In figure 7, we show the logit lens accuracies of GPT-J over the 8 datasets, and confirm that they are similar to the accuracies without calibration, albeit a bit noisier.

A.3 Logit lens results for the other models

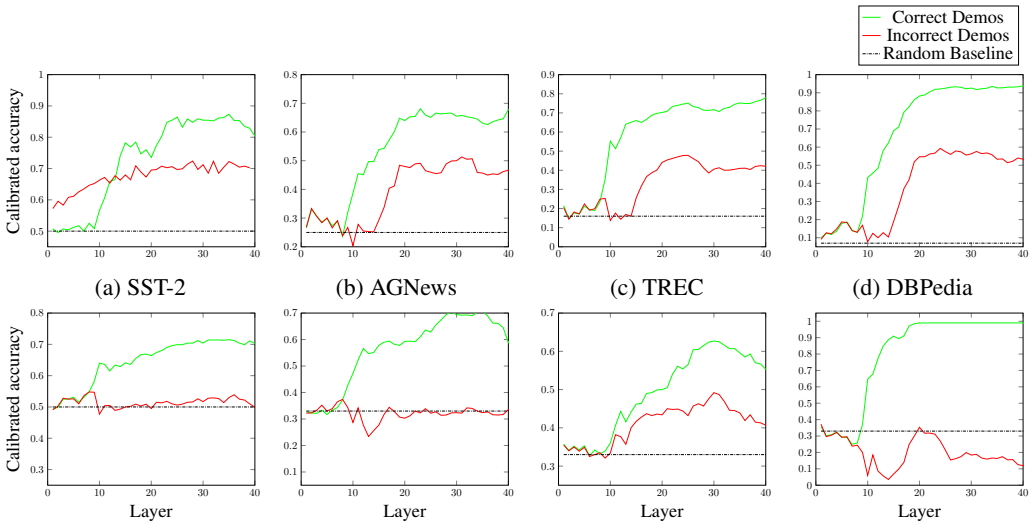
We plot the Logit Lens results for GPT2-XL and GPT-NeoX in Figure 8 and Figure 9.



(a) SST-2 (b) AGNews (c) TREC (d) DBPedia

(e) TE-Hate (f) SICK (g) PoemSentiment (h) Unnatural

Figure 8: GPT2-XL early-exit classification accuracies across 8 datasets, given correct and incorrect demonstrations. Given incorrect demonstrations, zeroing out all transformer blocks after layer 30 outperforms running the entire model on 7 out of 8 datasets.



(a) SST-2 (b) AGNews (c) TREC (d) DBPedia

(e) TE-Hate (f) SICK (g) PoemSentiment (h) Unnatural

Figure 9: GPT-NeoX early-exit classification accuracies across 8 datasets, given correct and incorrect demonstrations. Given incorrect demonstrations, zeroing out all transformer blocks after layer 23 outperforms running the entire model on 6 out of 8 datasets.