Surely You're Lying, Mr. Model: Improving and Analyzing CCS

Naomi Bashkansky [*1] **Chloe Loughridge** [*1] **Chuyue Tang** [*1 2]

Abstract

Contrast Consistent Search (Burns et al., 2022) is a method for eliciting latent knowledge without supervision. In this paper, we explore a few directions for improving CCS. We use conjunctive logic to make CCS fully unsupervised. We investigate which factors contribute to CCS's poor performance on autoregressive models. Replicating (Belrose & Mallen, 2023), we improve CCS's performance on autoregressive models and study the effect of multi-shot context. And we better characterize where CCS techniques add value by adding early exit baselines to the original CCS experiments, replicating (Halawi et al., 2023).[2]

1. Related Work

We want to find what the model thinks is the truth - that, is, its latent knowledge. To do so, we build upon work done by (Burns et al., 2022), who propose a method called Contrast Consistent Search (CCS) for eliciting the latent knowledge in language models by constructing a set of positive and negative prompts and training a probe to separate them.

However, using CCS for classification tasks requires giving the model an extra bit of information because while CCS can divide prompts into two clusters, it does not actually distinguish which prompt cluster is true and which cluster is false. We try not to rely on this extra information by using logical conjunction in section 2.1.

Concerningly, CCS fails in all autoregressive models. As an attempt to explain and fix the failure with autoregressive models like GPT-J, a very recent and as of yet unpublished work from Eleuther AI (Belrose & Mallen, 2023) suggests an improved loss function (VINC) to find a vector in the embedding space to separate contrastive prompts. We study factors that may contribute to the failure, explore methods based on VINC, and investigate the effects of truthful and corrupted context examples. 2.2.

As pointed out in a critiquing blog post (Roger, 2023), Burns et al. do not include random probe baselines for their CCS accuracy results. Roger provides evidence that CCS performs only marginally better than random linear probes on most of the datasets, and we corroborate this in section 2.3. Another helpful observation provided by Roger is that CCS probes trained on hidden states earlier in the GPT-J model are less sensitive to the format of the input prompt. To build upon this last observation, we reference research by (Halawi et al., 2023). Halawi et al. find that studying the early phases of computation in autoregressive transformer models like GPT-J could be a promising strategy for preventing the models from giving misleading (false) outputs. The study finds that ablating later layers in GPT-J reduces its tendency to follow false context in its prompt. It identifies certain attention heads that are responsible for paying attention to and replicating false context. Ablating these improves the accuracy of the model, even with misleading prompts. These insights justify the increased accuracy scores we observe from early exits in the earlier layers of GPT-J. 2.3

2. Our Contribution

We improve and explore CCS in three ways. This is an exploratory paper, and we hope that the results presented here contribute to an ongoing research process about how to best identify the development of models' "true beliefs".

- We make CCS fully unsupervised using conjunctions.
- We investigate factors that may contribute to CCS's poor performance with autoregressive models, use insights from (Belrose & Mallen, 2023) to improve the performance on an autoregressive model, and study the effect of multi-shot context on the results.
- We use insights from (Halawi et al., 2023) to provide a stronger zero-shot baseline for CCS results at *every* layer of the GPT-J model. This allows us to better qualify where CCS approaches offer marginal benefits over early exiting methods.

We perform all experiments with the GPT-J model (Wang &

[*]Equal contribution [1]Harvard University, Cambridge, MA [2]Peking University, Beijing, China. Correspondence to: Naomi Bashkansky <naomibashkansky@college.harvard.edu>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

[2]Anonymized code: `https://anonymous.4open.science/r/discovering_latent_knowledge-2148/README.md`

Komatsuzaki, 2021).

2.1. Use logical conjunctions to find the "truth" direction of the classifier

CCS splits the data into two clusters; however, it doesn't indicate which cluster is true and which is false. Burns et al. noted that there might be a way to distinguish between these two without using labels (Burns et al., 2022): by using natural language to form "and" and "or" statements; if [statement 1] and [statement 2] are in different clusters, and "[statement 1] & [statement 2]" falls into the same cluster as [statement 1], then [statement 1] must be in the "False" cluster (since "True & False" = "False").

The positive version of our prompts is formatted as in figure 2. The negative version of all prompts follows the same pattern, except we plug in the negative version of the sentiment label at the end of each prompt.

Doing PCA, we find that the hidden activations form 3 different clusters (Figure 1). Our decision of whether to flip the CCS predictions comes from the percent of "and" statements predicted to be True vs the percent of "or" statements predicted to be True; the former should be much smaller than the latter. If the former is greater than the latter, then we flip our predictions. This results in 70% accuracy on the amazon_polarity dataset, without ever having used the labels to make our decision! As the upper-bound baseline, Logistic Regression achieves a score of 85%.

2.2. Towards better eliciting latent knowledge on autoregressive models

CCS's poor performance on autoregressive models was pointed out in (Burns et al., 2022). This is worrying, since models like GPT-X have gained wide usage these days yet their latent knowledge is still unknown.

First, we look at several reasons that might cause the failure, including the sentence length and lack of context. Next, we seek other methods to find a proper direction vector in the embedding space of our contrast pairs and compare them with the CCS vector. We also test the robustness of new methods under corrupted contexts.

2.2.1. Possible effects leading to CCS failure with autoregressive models

We thought of two reasons that might result in CCS's failure with autoregressive models: 1) the length of the statement is too long; 2) it lacks a certain context. Some preliminary results show that these two are not the critical reasons.

Sentence Length We pick out two datasets, `amazon_polarity` and `sst2`. The average token length of the former dataset is about ten times the latter. For analysis, we randomly pick out 500 examples from each dataset, fit the CCS vector on 250 train examples, and get the accuracy on 250 test examples. Although reducing sentence length was helpful for the ground truth classifier obtained by linear regression (LR), the CCS performance degraded (Figure 3).

In-context Learning It's widely known that truthful multi-shot prompts can improve the model's performance. Since a longer sentence length does not significantly influence the results, we wonder whether adding ground truth context examples might help shape the embedding space and improve CCS vector performance. To test that, for each statement, we randomly pick $c = 10$ context sentences using the same format as the query statement. For example, we stick to one format `"Review:<>; Sentiment:<>"` and use the ground-truth answer for the context. However, the CCS vector performance degraded when we added more context sentences (Figure 4).

To better understand the difference embedding $X = X^+ - X^- \in \mathbb{R}^{N \times d}$, we project each feature onto a 2D plane via PCA. It seems that by adding more context sentences, it is harder to separate features (Figure 5, Figure 6).

2.2.2. Other directions in the embedding space that can help separate contrastive pairs

After several failed trials in improving the performance of CCS vector with autoregressive models, we turn to another idea: if we assume that the embeddings of contrast pairs contain critical information about truth, we should try other ways to analyze this embedding space.

For a dataset of contrast pairs $\{x_i^+, x_i^-\}_{i=1}^N$, we get the embedding $\{h(x_i^+), h(x_i^-)\}_{i=1}^N$ by passing them through the GPT-J model. We get the difference embedding $X \in \mathbb{R}^{N \times D}$ by $h(x^+) - h(x^-)$. $D = 4096$ is the embedding dimension of GPT-J. We formalize the problem as unsupervisedly finding a direction in the embedding space of contrast pairs. The score of each sample is the dot product between this direction vector $\mathbf{w}$ and the difference embedding $h(x_t^+) - h(x_t^-) \in \mathbb{R}^D$. Classify x_t by its score.

While we were working on this project, we noticed another ongoing project (Belrose & Mallen, 2023). One of their main ideas is *paraphrase invariance*. The understanding of truthfulness should stay consistent across various types of prompts of the same content. They hypothesize that explicitly regularizing a classifier's predictions to be invariant across data augmentation can improve performance. Throughout the text, we stay consistent with the original authors (Belrose & Mallen, 2023) and call this method VINC.

Contrast Negative Covariance A common idea is to find the top principal component (TPC) of the covariance matrix of the difference embedding matrix X as the direction.

$$\mathbf{w}^* = argmax_{\mathbf{w}:||\mathbf{w}||=1} \mathbf{w}^T \text{Cov}(X, X)\mathbf{w}$$

We could rewrite this as:

$$\begin{aligned} f_{covariance}(\mathbf{w}) &= \mathbf{w}^T \text{Cov}(X, X)\mathbf{w} && (1) \\ &= \mathbf{w}^T \text{Cov}(X^+ - X^-, X^+ - X^-)\mathbf{w} && (2) \\ &= \mathbf{w}^T [Cov(X^+, X^+) && (3) \\ &+ Cov(X^-, X^-) && (4) \\ &- 2Cov(X^+, X^-)]\mathbf{w} && (5) \end{aligned}$$

Intuitively, it can be understood as finding a direction along which X^+ and X^- are negatively correlated. Then, we compare the result between the direction given by CCS and the first principal component of the difference matrix X. However, this direction is almost as bad as the CCS direction (Figure 7).

Paraphrase Invariance According to (Belrose & Mallen, 2023), each statement x_i is augmented with k different kinds of prompts. Then, for each sample x_i, we can get embedding matrices $X_i^+, X_i^- \in \mathbb{R}^{k \times D}$. The key idea is that **the variance of projections onto the ideal direction among k prompts should be low**. Consider N statements, we have:

$$f_{invariance}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} -\mathbf{w}^T [Var(X_i^+) + Var(X_i^-)]\mathbf{w} \quad (6)$$

$$= \mathbf{w}^T [-\frac{1}{N}\sum_{i=1}^{N} Var(X_i^+) \quad (7)$$

$$- \frac{1}{N}\sum_{i=1}^{N} Var(X_i^-)]\mathbf{w} \quad (8)$$

Together with the above terms, we find the direction w^* by finding the top principal component of the inner combined matrix. By forcing a direction to be invariant across paraphrase prompts, we obtain a significant improvement (Figure 7).

$$\mathbf{w}^* =_{\mathbf{w}:||\mathbf{w}||=1} \alpha f_{covariance}(\mathbf{w}) + \beta f_{invariance}(\mathbf{w}) \quad (9)$$

Ablation Study To validate that the number of prompt variants matters and see how many variants are necessary, we run the same algorithm with different numbers of prompt variants k. The performance degrades when $k \leq 3$. Also, notice that when $k = 1$, this method is equivalent to the TPC method, and it is aligned with the previous fact that TPC is worse than CCS (Figure 8).

2.2.3. Robustness of truth discovery under corrupted contexts

Previously, we discuss whether **truthful** multi-shot contexts can help find a better CCS vector, and it turns out that the performance degrades. Now, since we have seen how VINC can find a perfect vector in the embedding space, we are curious whether VINC will be affected by **corrupted** multi-shot contexts. This is meaningful since when people are prompting with some wrong contexts, we still hope to see that the model contains the truthful information about the last query statement.

Specifically, we wonder:

1. whether adding contexts could help VINC vector in gaining better accuracy;
2. whether VINC is robust under different levels of context corruption.

As for 1), we compare the accuracy between without context and with different numbers of context examples and it turns out that the performance consistently increases when more examples are added (Figure 9). As for 2), we use the number of context examples $c = 10$ and corrupt them with probability p. By corruption, we specifically mean that we flip the answer. In dataset `amazon_polarity`, this would mean to fill in the answer block with `positive` when the correct answer is `negative` and vice versa. The performance of VINC vector is really robust against all levels of corruption (Figure 10).

2.3. Another Baseline: Extension using LogitLens

The CCS paper by Burns et al. compares the accuracy of the CCS method on GPT-J's last layer to the zero-shot performance of GPT-J on the test set. However, Burns et al. do not investigate how the accuracy of zero-shot prediction based on the logit scores from intermediate model layers (an approached dubbed "early exiting" by the "Overthinking the Truth" paper) compares with the accuracy of CCS probes trained on those intermediate layers (Burns et al., 2022). The "Overthinking the Truth" paper gives as reason to believe that early exiting may give more competitive results to CCS on intermediate layers (Halawi et al., 2023).

2.3.1. Methods for performing early exiting analysis

In order to extract the model's intermediate beliefs about the answer to the prompt, we use the LogitLens approach first introduced by (nostalgebraist, 2020) and further applied (Halawi et al., 2023).

First, we extract the hidden states from each intermediate layer, resulting in a tensor of shape (N, D, L). We use $N = 250$ examples, $D = 4096$ as GPT-J's embedding dimension, and $L = 29$ as the number of layers in GPT-J. Following previous section, we also study the multi-shot setting, where $c = 0, 1, 5, 10$. The intermediate logit predictions at layer l for test sample i can be extracted using equation 10.

$$\text{Logits}_i^l = [\text{logit}_0^l, ..., \text{logit}_{|V|}^l] = W_U \cdot \text{LayerNorm}_L(\text{hidden}_i^l) \tag{10}$$

Here, W_U is GPT-J's unembedding matrix (which is in fact the same as GPT-J's embedding matrix) and is of size $|V| \times D$ where $|V|$ is the length of GPT-J's vocabulary. L stands in for the total number of layers in the model, and importantly we pass the hidden state for every layer through GPT-J's *final* LayerNorm layer before multiplying by the unembedding matrix.

Once we have the logits for a given layer, we can find the difference between the "positive" logit and the "negative" logit for sample i at layer l as follows:

$$\text{logit\_diff}_i^l = \text{Logits}_i^l[\text{"positive"}] - \text{Logits}_i^l[\text{"negative"}] \tag{11}$$

We visualize the average logit differences across all layers of GPT-J in figure 2 with the zero-context prompt (the logit difference for each layer has been averaged over the 250 samples from our amazon_polarity test set). Interestingly, as the context length increases, the difference between positive and negative logits markedly increases as a peak between layers 0 and 10 of the model.

Next, we'd like to turn these logit scores and logit differences into intermediate predictions from the model. We explore four approaches for doing this. First, we attempt an unnormalized approach: for each sample, if the logit difference is positive we predict that the review is positive, and if the logit difference is negative, we predict that the review is negative. Often, however, for a given layer, the model outputs either "positive" or "negative" for *all* samples, revealing a strong bias. We attempt to normalize against this bias using two strategies.

We call the first normalization strategy the Mean Difference normalization strategy. We output "positive" for a sample if the logit difference for that sample is greater than the median logit difference across all the samples for that layer. This works even if all logit differences are negative.

We call the second normalization strategy the Median Posval strategy. Under this normalization scheme, we output "positive" for a sample if the logit score for the "positive" token is greater than the median "positive" token logit score across all samples. This ensures the model outputs 50% positive predictions and 50% negative predictions for the test set. We note this is effectively the same normalization strategy used by Steinhardt et al. in "Overthinking the Truth" (Halawi et al., 2023).

2.3.2. Effects of different normalization schemes on early exiting

The Median Difference and Median Posval normalization strategies for GPT-J early-exit predictions yield largely similar accuracy results, but both provide more nuanced insight into the accuracy changes between predictions from different layers of the model. Based on this set of empirical results, the choice of Median Difference normalization versus Median Posval normalization shouldn't matter. Please see 1 for empirical evidence.

Encouragingly, the normalized early-exit accuracies for helpful prompts agree with results from (Halawi et al., 2023). A subtle but clear pattern emerges in which layers 5-13 of GPT-J produce predictions with the highest accuracy scores, which barely beat the random baseline and sometimes beat CCS. Across 16 other datasets, (Halawi et al., 2023) noticed this same localized pattern in GPT-J. We're happy to provide another data set and corroborating evidence.

2.3.3. Discussion

The impacts of these results are somewhat limited, because CCS is demonstratedly bad for autoregressive models like GPT-J. Additionally, we arbitrarily chose the dataset amazon_polarity, and to make any strong conclusions we would need to ideally replicate these results across many datasets with many autoregressive models. However, these results do perhaps offer a qualification to the benefits of the CCS method: when applied to GPT-J, it mainly provides marginal value in the later layers, when the model supposedly begins paying more attention to misleading prompts (as demonstrated by (Halawi et al., 2023)).

2.3.4. Future research directions

In the future, it would be ideal to compare the robustness of early exit prediction accuracies on increasingly corrupt prompts with the robustness of CCS accuracies on increasingly corrupt prompts. We didn't immediately pursue this because CCS already performs quite poorly on helpful prompts (barely above the random baseline in most instances). However, upon further reflection, we may still pick up on some robustness signal if we do proceed with these tests. Furthermore, it would be interesting to extend these analyses to a wider range of model sizes to see if the patterns between layers 5 and layers 13 of GPT-J extend to larger models as well. Finally, it could be helpful to investigate if any insights from the refined TunedLens paper (Belrose et al., 2023) might be applicable to our usecase as a substitute for the original LogitLens approach.

References

Belrose, N. and Mallen, A. Eliciting latent knowledge. https://github.com/EleutherAI/elk, 2023.

Belrose, N. et al. Eliciting latent predictions from transformers with the tuned lens. https://arxiv.org/pdf/2303.08112.pdf, 2023.

Burns, C. et al. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Halawi, D., Denain, J.-S., and Steinhardt, J. Overthinking the truth: Understanding how language models process false demonstrations. 2023.

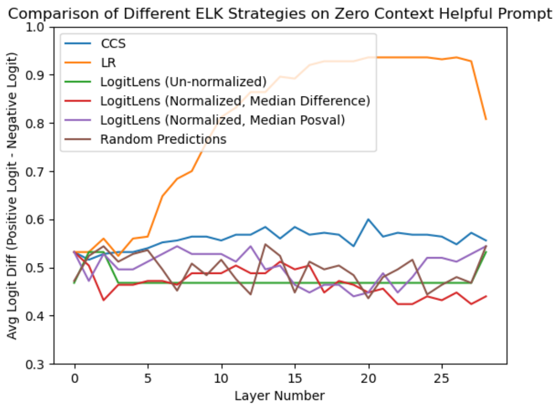
nostalgebraist. Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020.

Roger, F. What discovering latent knowledge did and did not find. *LessWrong*, 2023.

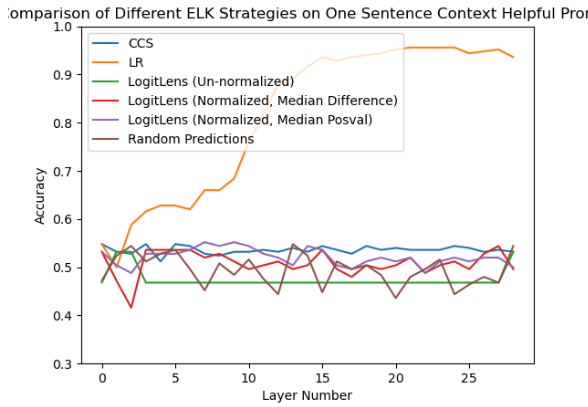
Wang, B. and Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

A. Appendix

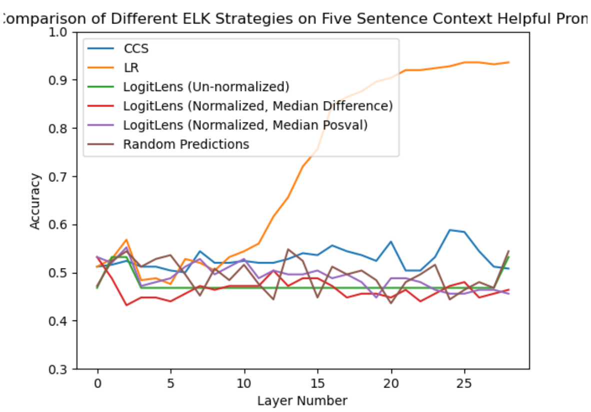
Table 1. Early Exit Accuracies for Prompts with Different Helpful Context Lengths



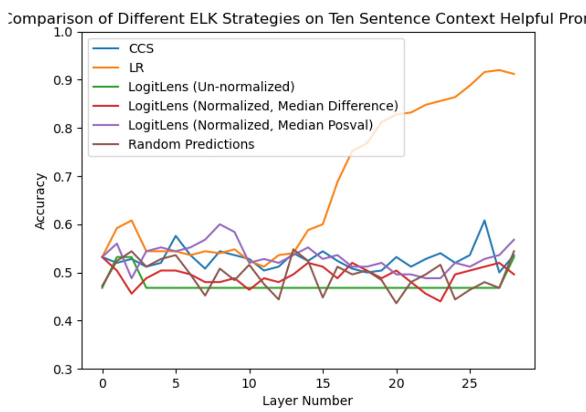
figureEarly Exit Accuracy on Zero Context Helpful Prompts



figureEarly Exit Accuracy on One Sentence Context Helpful Prompts

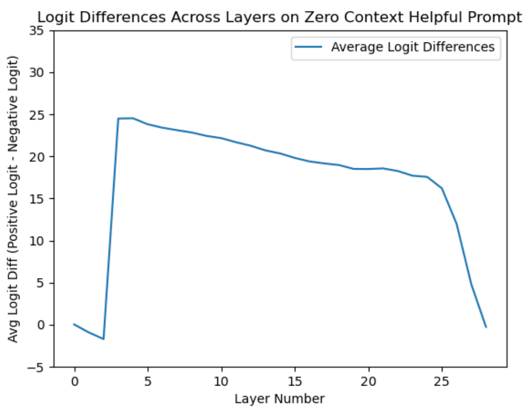


figureEarly Exit Accuracy on Five Sentence Context Helpful Prompts

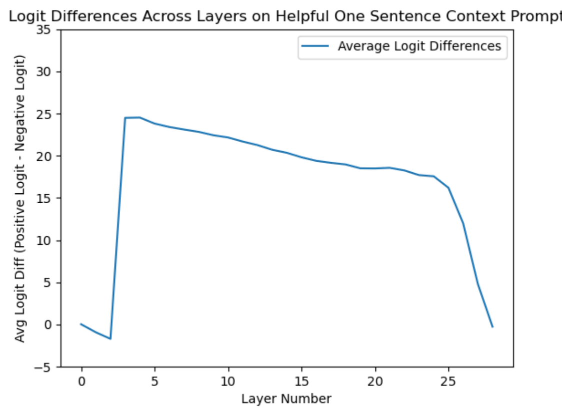


figureEarly Exit Accuracy on Ten Sentence Context Helpful Prompts

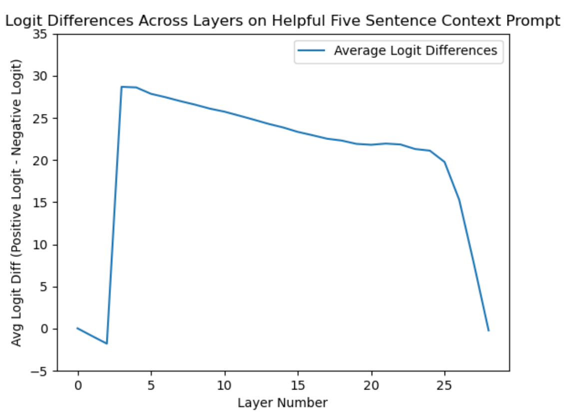
Table 2. Average Logit Differences for Prompts with Different Helpful Context Lengths



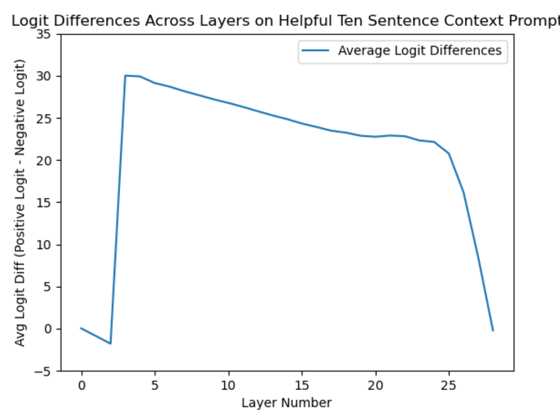
figureAvg Logit Diffs on Zero Context Helpful Prompts



figureAvg Logit Diffs on One Sentence Context Helpful Prompts



figureAvg Logit Diffs on Five Sentence Context Helpful Prompts



figureAvg Logit Diffs on Ten Sentence Context Helpful Prompts

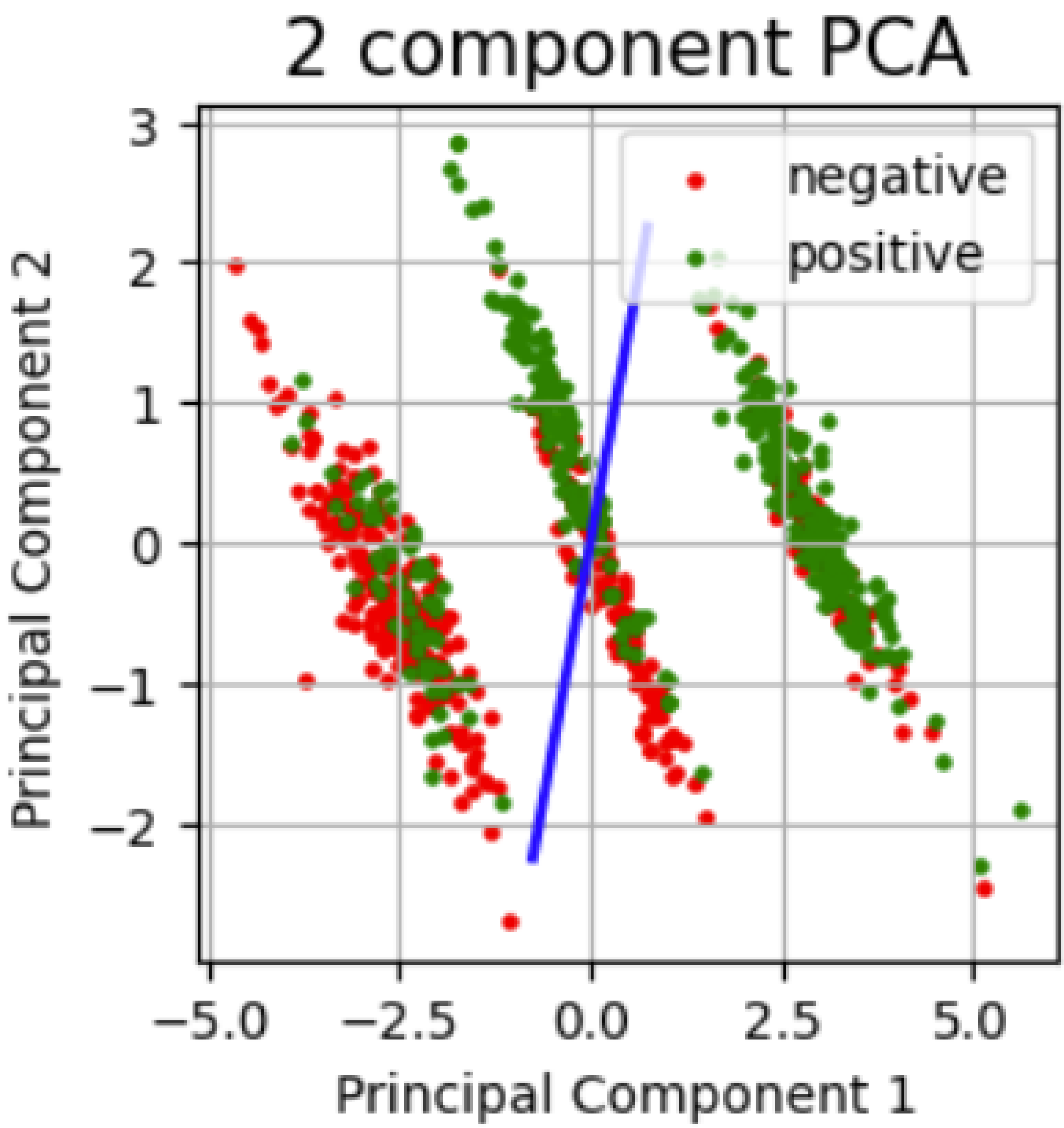


Figure 1. In this PCA plot, the green points are the ground-truth positives, the red points are the ground-truth negative and the blue line in the Logistic Regression decision boundary.

Positive Prompt (truth: positive)
"This was a easy to read book that made me want to keep reading on and on, not easy to put down.It left me wanting to read the follow on, which I hope is coming soon. I used to read a lot but have gotten away from it. This book made me want to read again. Very enjoyable."
Between negative and positive, the sentiment of this example is **positive**.

"A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up to absolutely nothing. I'm embarrassed for this author and very disappointed I actually paid for this book."

Positive AND Prompt (truth: negative)
"This was a easy to read book that made me want to keep reading on and on, not easy to put down.It left me wanting to read the follow on, which I hope is coming soon. I used to read a lot but have gotten away from it. This book made me want to read again. Very enjoyable. A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up to absolutely nothing. I'm embarrassed for this author and very disappointed I actually paid for this book."
Between partly negative and completely positive, the sentiment of this example is **completely positive**.

Positive OR Prompt (truth: positive)
"This was a easy to read book that made me want to keep reading on and on, not easy to put down.It left me wanting to read the follow on, which I hope is coming soon. I used to read a lot but have gotten away from it. This book made me want to read again. Very enjoyable. A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up to absolutely nothing. I'm embarrassed for this author and very disappointed I actually paid for this book."

Between completely negative and partly positive, the sentiment of this example is **partly positive**.

Figure 2. Examples of the positive form of normal, AND, and OR prompts.

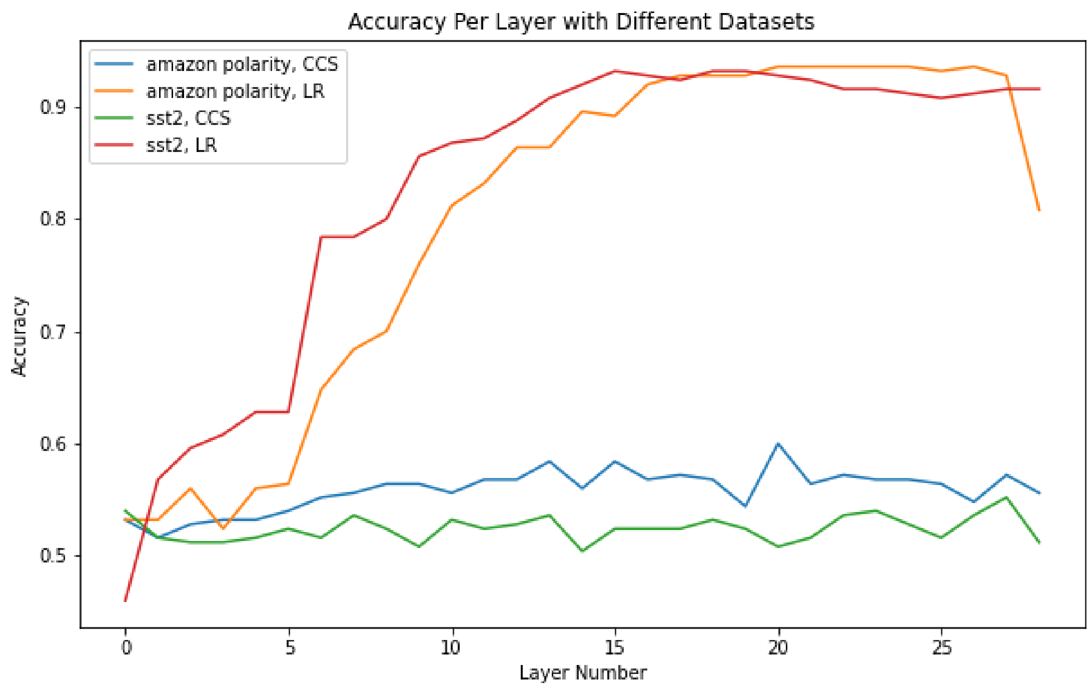


Figure 3. Compare the CCS and ground truth performance across layers between shorter dataset `sst2` and longer dataset `amazon_polarity`.

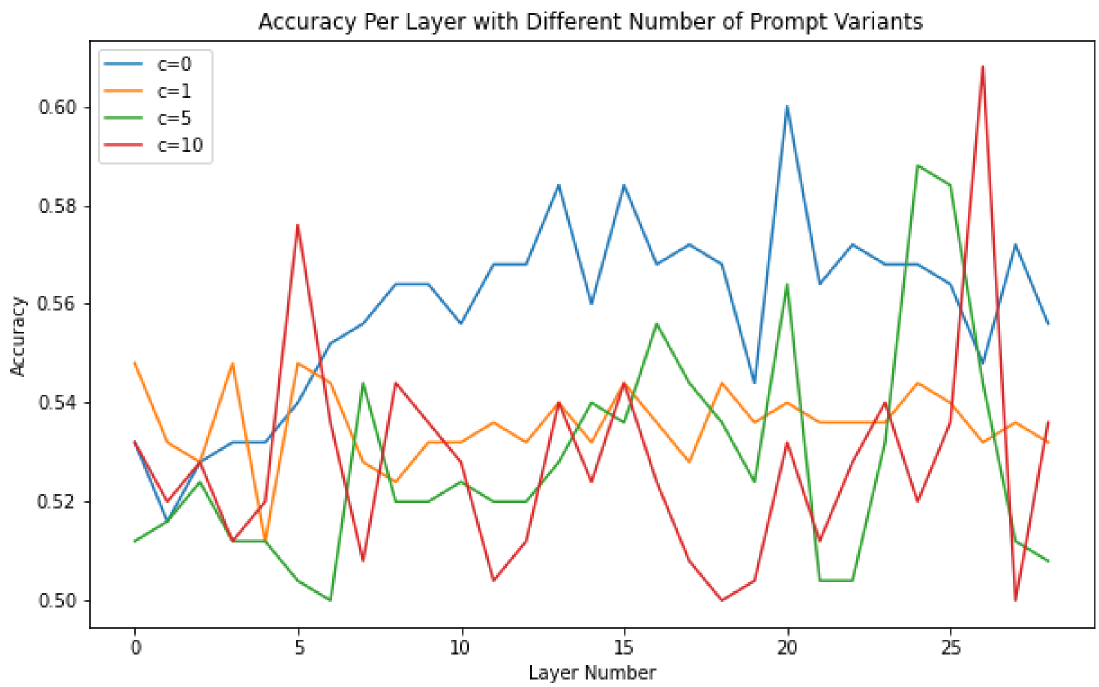


Figure 4. Compare the CCS performance across layers with different numbers of context examples. Adding context examples reduce the accuracy.

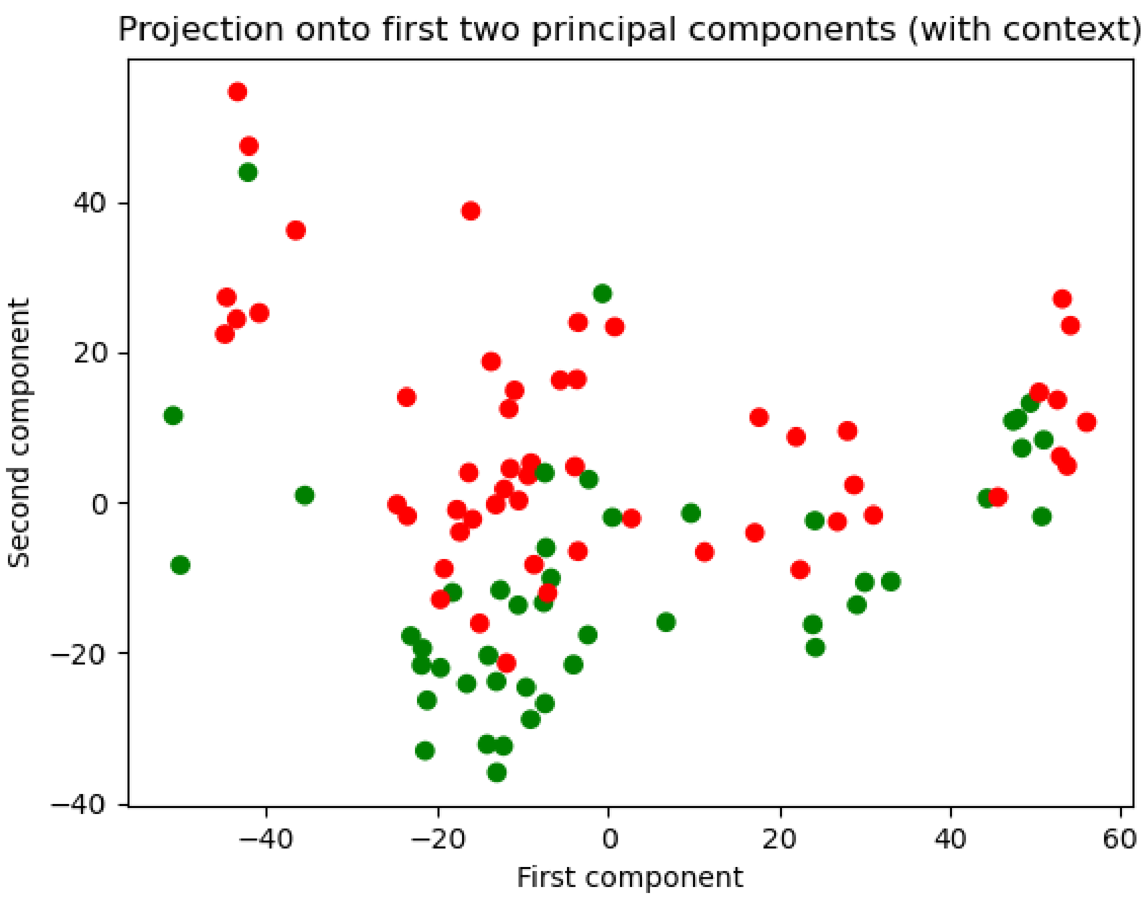


Figure 5. With 10 context sentences

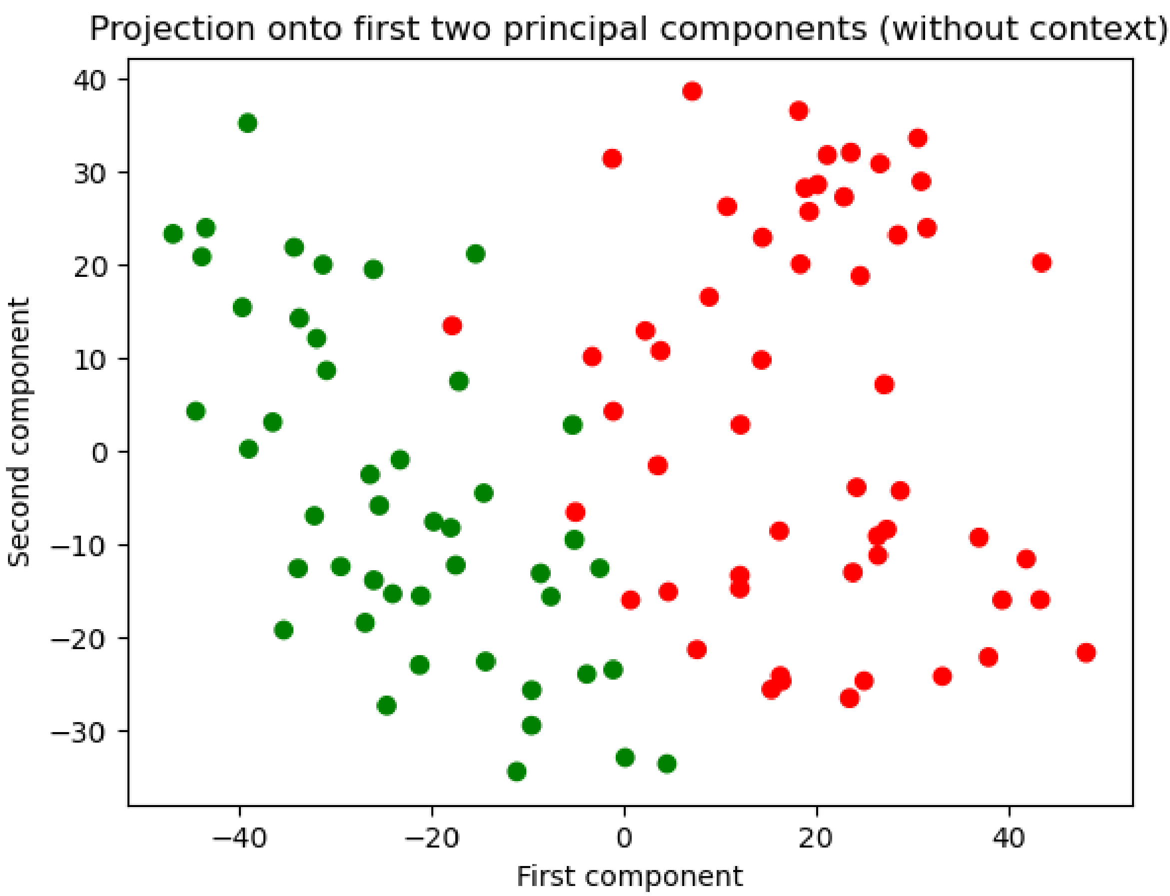


Figure 6. Projecting the contrast embeddings onto the 2D plane expanded by the top two principal components. Red dots refer to label=0 and green dots refer to label=1.

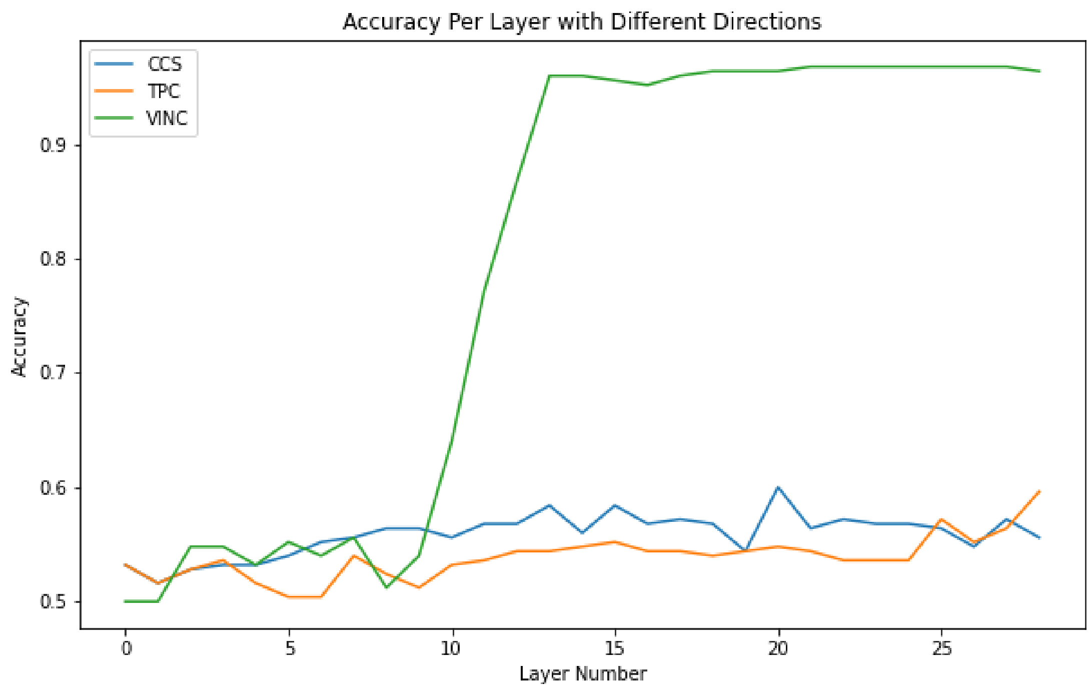


Figure 7. Compare the performance of three ways of finding directions in the embedding space across layers, CCS, TPC, and VINC.

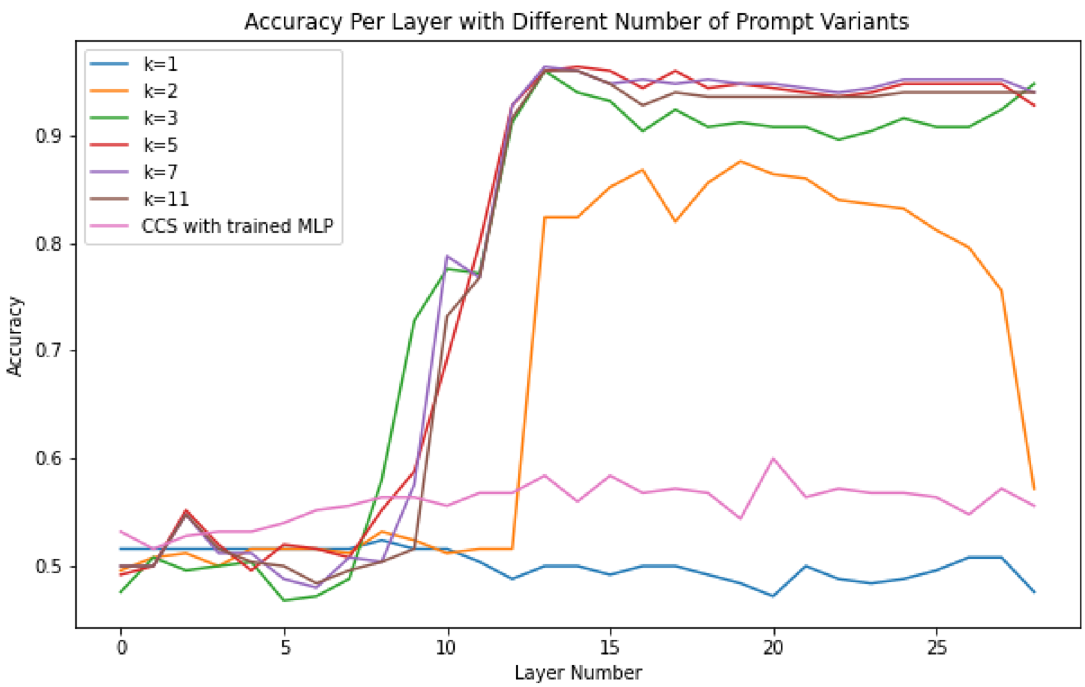


Figure 8. Ablation on paraphrase invariance in VINC. When the number of prompt variants k reduces to ≤ 3, performance drops. When $k = 1$, VINC degrades to TPC.

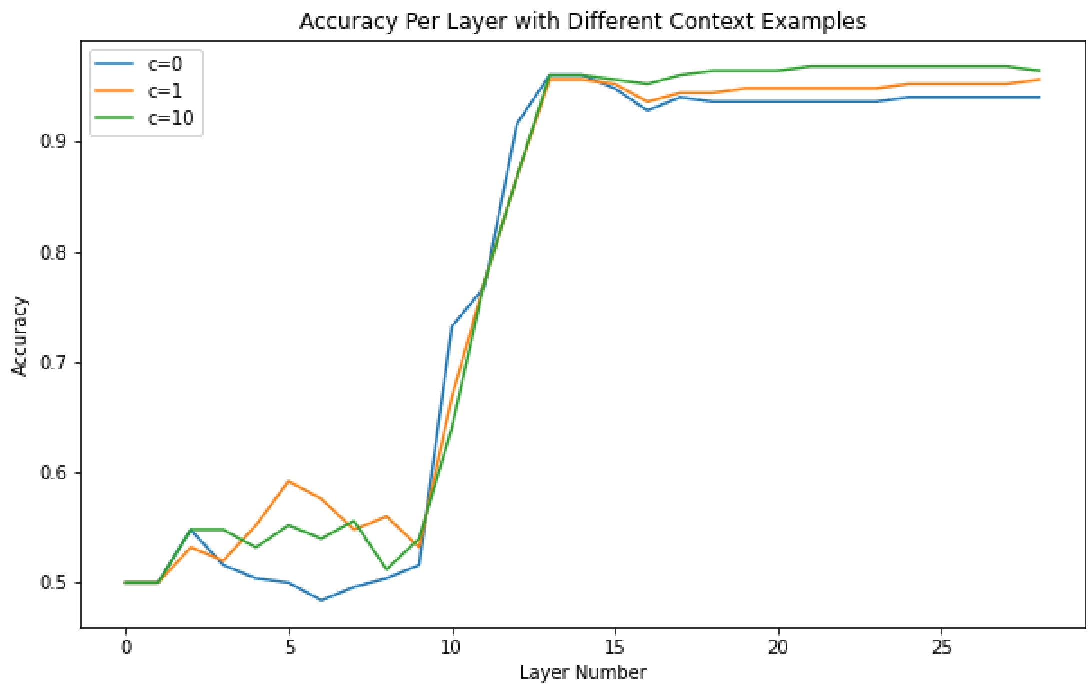


Figure 9. Compare the performance of VINC under different numbers of truthful context examples $c = 0, 1, 10$. The performance is consistently improved when more context examples are added.

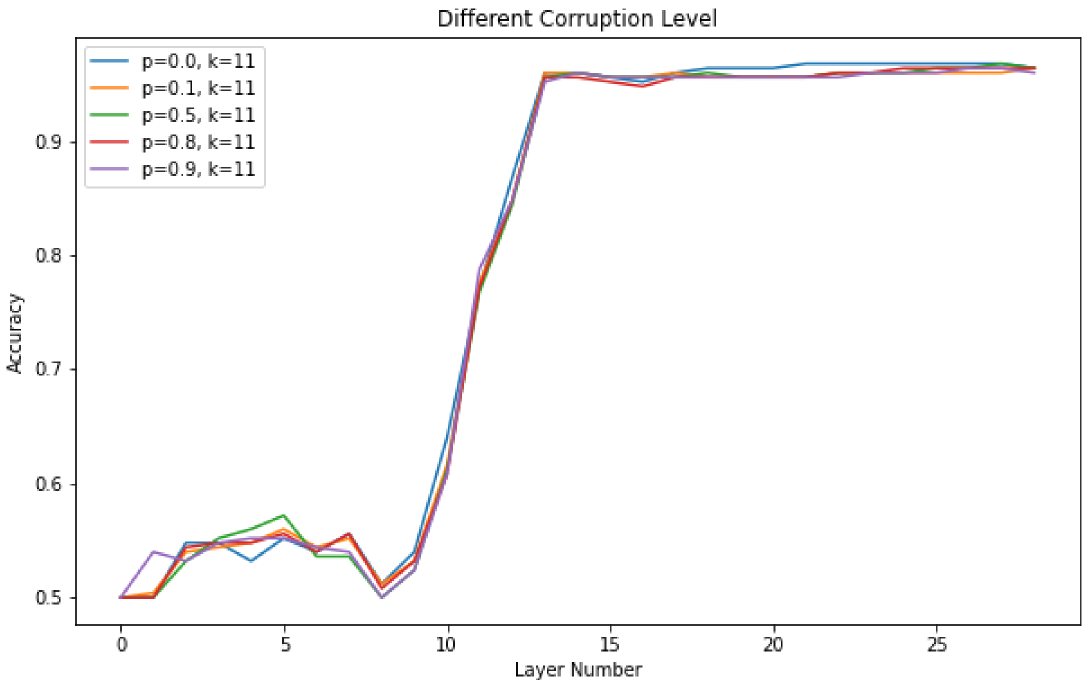


Figure 10. Corrupt $c = 10$ context examples with different corruption probability $p = 0.0, 0.1, 0.5, 0.8, 0.9$. No corruption case only slightly outperforms the corruption cases.

Zero-shot context

Review: Check out Maha Energy's website. Their Powerex MH-C204F charger works in 100 minutes for rapid charge, with option for slower charge (better for batteries). And they have 2200 mAh batteries.
Sentiment:

One sentence helpful context

Review: 8 Crazy Nights might have been a sweet film with a good message for kids, but the scatological humor, offensive language and explicit sexual references made it unsuitable for my 10-year old. The plot, on the other hand, while fine for 10-year olds was too obvious and simplistic for most of the adults in the audience. As a result, while it's probably not the worst film of the year, it is certainly in the running.
Sentiment: negative
Review: Check out Maha Energy's website. Their Powerex MH-C204F charger works in 100 minutes for rapid charge, with option for slower charge (better for batteries). And they have 2200 mAh batteries.
Sentiment:

We extend to five and also ten sentence helpful contexts.

Figure 11. Prompts used for early exiting analyses.