Cluster-Norm for Unsupervised Probing of Knowledge

Walter Laurito[1,2,*], **Sharan Maiya**[2,3,*], **Grégoire Dhimoïla**[4,2,*],
Ho Wan Yeung[5], **Kaarel Hänni**[6]

[1]FZI, [2]Cadenza Labs, [3]University of Cambridge, [4]ENS Paris-Saclay, [5]UC Berkeley, [6]Caltech

Correspondence: laurito@fzi.de

Abstract

The deployment of language models brings challenges in generating reliable information, especially when these models are fine-tuned using human preferences. To extract encoded knowledge without (potentially) biased human labels, unsupervised probing techniques like Contrast-Consistent Search (CCS) have been developed (Burns et al., 2022). However, salient but unrelated features in a given dataset can mislead these probes (Farquhar et al., 2023). Addressing this, we propose a cluster normalization method to minimize the impact of such features by clustering and normalizing activations of contrast pairs before applying unsupervised probing techniques. While this approach does not address the issue of differentiating between knowledge in general and simulated knowledge—a major issue in the literature of latent knowledge elicitation (Christiano et al., 2021)—it significantly improves the ability of unsupervised probes to identify the intended knowledge amidst distractions.[1].

1 Introduction

The deployment of language models for practical applications introduces novel challenges, including the potential creation of untrustworthy or incorrect text (Weidinger et al., 2021; Park et al., 2023; Evans et al., 2021; Hendrycks et al., 2021). Specifically, models that are fine-tuned using human preferences may amplify existing human biases or generate persuasive yet deceptive outputs (Perez et al., 2022).

Empirical evidence suggests that simulated internal beliefs or *knowledge* can be extracted from language model activations (Li et al., 2022; Gurnee and Tegmark, 2023; Azaria and Mitchell, 2023; Bubeck et al., 2023). Supervised probing methods can be employed to extract this knowledge (Alain and Bengio, 2016; Marks and Tegmark, 2023) but such methods require labels, which in some domains may not be readily provided due to human biases or because humans simply do not know the correct label. It may even be critical to avoid the use of human labels to differentiate between a model's true knowledge and its representation of human knowledge. Motivated by these ideas, unsupervised probing techniques like Contrast-Consistent Search (CCS) have been developed to extract the knowledge embedded in a language model without the need for ground truth labels (Zou et al., 2023; Burns et al., 2022).

Farquhar et al. (2023) outline current limitations of these approaches, demonstrating that these unsupervised probes tend to identify the most salient binary feature, which may not always correspond to the specific knowledge feature we seek. For example, in one experiment, one of a pair of distracting random words is added to each prompt in a text dataset. After training, unsupervised CCS probes often function as classifiers for these random words, rather than the intended knowledge feature. In practice, there may be numerous salient features of which we are unaware, which can divert an unsupervised probe from identifying the target feature, regardless of whether they are correlated or uncorrelated with the target.

To tackle this issue, we propose a cluster normalization method. Our method starts by following the usual initial approach of unsupervised probing of harvesting contrast pair activations. We then cluster similar activations and normalize them separately, thereby eliminating the effect of distracting salient features. We can then apply any unsupervised probing method, such as CCS or CRC-TPC (Burns et al., 2022), to train a probe on these normalized activations. It is of course crucial to ensure this approach does not inadvertently eliminate the knowledge feature itself. To prevent this, we utilize contrast pairs, performing the clustering on the average embedding of each pair. Further details on

[*]These authors contributed equally to this work.

[1]The code for this work is available at https://github.com/Cadenza-Labs/cluster-normalization.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14083–14112
November 12-16, 2024

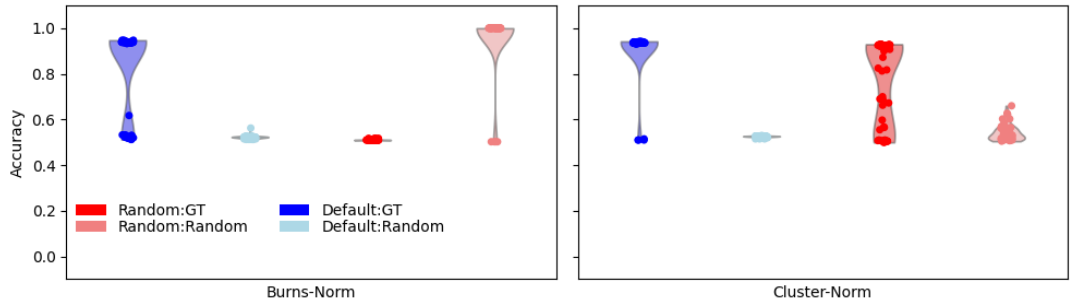

Figure 1: Under the standard CCS approach (left; Burns-Norm), a modified prompt including distracting random words (red) causes all CCS probes to achieve random accuracy against ground truth labels (GT), and high accuracy against these random word labels (random). When using our approach of cluster normalization (right; Cluster-Norm), the average accuracy of CCS probes for the desired feature increases significantly.

contrast pairs are provided in Section 2.1.

Probes trained with the original CCS approach achieve an average accuracy of approximately 0.5 on prompt datasets with distracting random word features (Farquhar et al., 2023). In contrast, our clustering method significantly improves this average accuracy to about 0.77, and to 0.81 for CRC-TPC when tested with Mistral-7B. Details of our method are described in Section 3.

2 Background

2.1 Contrast-Consistent Search (CCS)

Contrast-Consistent Search (CCS), as described by Burns et al. (2022), locates a direction in activation space using a perceptron that adheres to logical consistency principles. This is achieved through a loss function designed to ensure that probabilities for a question-answer pair and its negated counterpart — a contrast pair — are complementary. This loss function is optimized in an unsupervised manner, and in doing so CCS extracts the latent knowledge within large language models to answer binary questions.

At first, a language model $\mathcal{M}$ processes a dataset of textual contrast pairs $(x_i^+, x_i^-)_{i=1}^n$, generating contextualized embeddings $(\mathcal{M}(x_i^+), \mathcal{M}(x_i^-))$. Following this, a linear probe (Alain and Bengio, 2016) is trained to calculate from these embeddings the probabilities p^+ and p^-, whether x_i^+ or x_i^- is true, respectively. The objective function used to train this probe is given by a sum of two terms:

$$\mathcal{L}_{\text{CCS}} = \sum_{i=1}^{N} \mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{confidence}}$$

$$\mathcal{L}_{\text{consistency}} = \left[p\left(x_i^+\right) - \left(1 - p\left(x_i^-\right)\right)\right]^2$$

$$\mathcal{L}_{\text{confidence}} = \min\left\{p\left(x_i^+\right), p\left(x_i^-\right)\right\}^2.$$

The first term, $L_{\text{consistency}}$, is motivated by the idea that the probabilities of a statement and its negation should sum to one. This ensures logical consistency. The second term, $L_{\text{confidence}}$, is designed to maximize the information extracted by the probe, penalizing cases where the probabilities for both true and false are the same, at $p(x^+) = p(x^-) = 0.5$. Thus, this term encourages the probe to be more certain in its outputs.

Intuitively, there are at least two possible directions (features) satisfying this loss. The first is the knowledge direction we seek, $\vec{F}_{\top/\perp}$, and the second is the syntactical difference between positive and negative prompt templates, $\vec{F}_{\pm}$. To remove this latter undesired feature, Burns et al. (2022) proceed as follows. Before training an unsupervised probe, contrast pair activations are first normalized: $\widetilde{\mathcal{M}}(x_i^+) = \frac{\mathcal{M}(x_i^+) - \mu^+}{\sigma^+}$, with μ^+ and σ^+ the mean and standard deviation of the activations of all positive examples in each contrast pair; the same normalization procedure is followed for negative examples, and the unsupervised probe is trained on these normalized $\widetilde{\mathcal{M}}(x_i^{\pm})$. In this way, CCS removes the most salient feature of the contrast pair differences: the syntactical difference direction $\vec{F}_{\pm}$. However, as Farquhar et al. (2023) show, the second-most salient feature may not necessarily be the desired knowledge - an implicit assumption of the original CCS method. In this work, we take advantage of normalization to remove other undesired salient features by including a clustering step.

2.2 Contrastive Representation Clustering

As an alternative to CCS, the method of *Contrastive Representation Clustering via Top Principal Component* (CRC-TPC) (Burns et al., 2022) separates the normalized contrast pair differences

$\{\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-)\}$ based on projections onto their top principal component, i.e., the singular vector associated with the highest singular value, or the direction with the highest variance. This is again motivated by the intuition that the most salient contrastive feature after $\vec{F}_\pm$ - removed by normalization - should be the knowledge feature $\vec{F}_{\top/\perp}$.

2.3 Theoretical Background

A *salient feature* is a direction with high variance in the data. We are interested in salient features in the contrast pair differences $\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-)$, and we refer to these as *contrastive features*.

In this section, we explain (1) why undesired salient contrastive features can mislead unsupervised probes, and (2) how contrastive features can be induced by non-contrastive ones. The mechanisms of the latter point are illustrated through an example.

We shall first examine why there is a close link between the CCS loss described in Section 2.1 and the idea of saliency i.e., variance. Contrastive features will naturally achieve a low CCS loss. To see this, consider the variance of contrast pair differences projected along the feature direction of a given feature $\vec{F}$:

$$X := \vec{F}^T \cdot \widetilde{\mathcal{M}}(x_i^+),\ Y := \vec{F}^T \cdot \widetilde{\mathcal{M}}(x_i^-).$$
$$Var(X-Y) = \underbrace{E(X^2) + E(Y^2)}_{Confidence} \underbrace{-2 \cdot E(X \cdot Y)}_{Consistency}$$
$$\underbrace{-(E(X)^2 + E(Y)^2 - 2 \cdot E(X) \cdot E(Y))}_{=0}$$

In this expanded form, we see that the variance of contrast pair differences in the direction $\vec{F}$ captures,

- **confidence**, with $E(X^2) + E(Y^2)$ higher if the magnitude of the projection along $\vec{F}$ in either element of a pair is high,
- and **consistency**, with $-2 \cdot E(X \cdot Y) > 0$ if the projections of a contrast pair along $\vec{F}$ have opposing sign. This also increases with the magnitude of these projections.

Note that the term $-(E(X)^2 + E(Y)^2 - 2 \cdot E(X) \cdot E(Y))$ equals zero under the set-up of CCS, as the normalization step described above results in $E(\widetilde{\mathcal{M}}(x_i^\pm)) = 0$, therefore $E(X) = E(Y) = 0$.

Due to this link between confidence, consistency, and contrastive saliency, a probe trained using the CCS loss will favor learning salient contrastive features. Otherwise, the projections of a contrast pair onto a feature will be small in difference or equal, failing to satisfy at least the consistency condition.

As mentioned in Section 2.1, we can describe two features which intuitively will satisfy the CCS loss:

- $\vec{F}_\pm := \vec{F}_+ - \vec{F}_-$, the syntactical difference between contrast pairs due to the appending of positive and negative tokens, removed by normalization in the original CCS method,
- $\vec{F}_{\top/\perp} := \vec{F}_\top - \vec{F}_\perp$, the knowledge feature we seek.

Under our definition, undesired distracting features such as proxies for knowledge or random words (as in Farquhar et al. (2023)) should not be contrastive features: their contrast pair projections should be equal in both examples of each pair and thus should be ignored by a CCS probe. It is however the case that these non-contrastive features can still mislead unsupervised probes by *inducing* undesired contrastive features. We describe the mechanism through which this occurs with the following example:

Let f be some binary function, say the XOR function on the presence of features, and $\vec{F}_1$, $\vec{F}_2$ be any two features. Suppose that a model represents the feature $f(\vec{F}_1, \vec{F}_2)$ as its own direction $\vec{F}_{f(\vec{F}_1,\vec{F}_2)}$, orthogonal to $\vec{F}_{1,2}$[2]. Now, fix two features $\vec{F}_1$ and $\vec{F}_2$ and assume without loss of generality that exactly one of them appears in each pair with probability $\frac{1}{2}$.

We can now write the contrast pair differences as:

$$\mathcal{M}(x_i^+) - \mathcal{M}(x_i^-) = \underbrace{\vec{F}_+ - \vec{F}_-}_{\vec{F}_\pm} + \underbrace{\vec{F}_{f(\vec{F}_+,\vec{F}_j)} - \vec{F}_{f(\vec{F}_-,\vec{F}_j)}}_{\Delta^\pm_{\vec{F}_j}}$$
$$\pm \underbrace{(\vec{F}_\top - \vec{F}_\perp)}_{\vec{F}_{\top/\perp}} \pm \underbrace{(\vec{F}_{f(\vec{F}_\top,\vec{F}_j)} - \vec{F}_{f(\vec{F}_\perp,\vec{F}_j)})}_{\Delta^\top_{\vec{F}_j}}$$

for $j \in \{1, 2\}$.

The expected value of these contrast pair differences over our dataset is:

$$E(\mathcal{M}(x_i^+) - \mathcal{M}(x_i^-)) = \vec{F}_\pm + \frac{1}{2}(\Delta^\pm_{\vec{F}_1} + \Delta^\pm_{\vec{F}_2}) \quad (1)$$

since $\vec{F}_\pm$ is constant, $\Delta_{\vec{F}_j}$ are both constant on half of the dataset and the two knowledge related terms

[2]Evidence of such behavior has been observed with $f = XOR$ and any two features, as shown in (Marks, 2024).

have uniformly alternating sign. After centering, we have:

$$\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-) = \pm \vec{F}_{\top/\perp} \pm \Delta^{\top}_{\vec{F}_j} + \alpha \frac{1}{2}(\Delta^{\pm}_{\vec{F}_1} - \Delta^{\pm}_{\vec{F}_2})$$

where $\alpha = 1$ if $j = 1$ and -1 otherwise.

A probe using any of these remaining terms will have low CCS loss, with a bias towards the most salient terms. This is true for any undesirable feature that would remain in the contrast differences, and it affects both trained CCS probes and analytical CRC-TPC probes. This work aims to address this issue by removing unwanted features *before* training the probe.

3 Method

We begin with a dataset of contrast pairs, $\{(x_i^+, x_i^-)\}_{i=1}^n$. For each pair, we harvest the intermediate activations of a language model $\mathcal{M}$, specifically the state of the residual stream at the final token position at a specific layer, which we denote as $\mathcal{M}(x_i^{\pm})$. We average these activations for each contrast pair $\mathcal{M}(x_i) = \frac{\mathcal{M}(x_i^+)+\mathcal{M}(x_i^-)}{2}$, and partition $\{\mathcal{M}(x_i)\}_i$ using a clustering algorithm, thereby partitioning the original dataset using its most salient features. Each cluster is then normalized separately to have zero mean and unit variance, i.e. for each positive sample x_i^+, where x_i belongs to cluster c, $\widetilde{\mathcal{M}}(x_i^+) = \frac{\mathcal{M}(x_i^+)-\mu_c^+}{\sigma_c^+}$, where μ_c^+ and σ_c^+ are the mean and standard deviation of all positive samples in cluster c. The same normalization process is applied to all negative samples. Finally, an unsupervised probe can be trained on the contrast pair differences of the normalized (by cluster) samples. This approach allows the probe to isolate the desired knowledge feature, ignoring other distracting features isolated to each original cluster.

Following the notation in Section 2.3, if x_i belongs to cluster $c \in \{1, 2\}$, a successful cluster normalization will leave:

$$\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-) = \pm\vec{F}_{\top/\perp} \pm \Delta^{\top}_{\vec{F}_c}.$$

This follows from equation 1, however in this case normalization is performed over c only as opposed to the whole dataset.

A key element to the effectiveness of our method is that our clustering approach does not erase the effect of the desired knowledge feature. This is achieved by clustering the averages of each contrast pair, $\mathcal{M}(x_i)$. As a result, clustering only isolates salient non-contrastive features, and is effectively blind to $\vec{F}_{\pm}$ and $\vec{F}_{\top/\perp}$. Normalizing positive and negative samples separately per cluster aims to ensure that all contrastive features $\vec{F}'$ related to $\vec{F}_{\pm}$ are properly normalized out - including the leaks from non-contrastive features $\vec{F}$ mixing with $\vec{F}_{\pm}$, as explained in Section 2.3. Note, we do not normalize out similar $\vec{F}'$ resulting from the mixing of $\vec{F}$ with $\vec{F}_{\top/\perp}$. Eventually, only contrastive features related to knowledge are kept.

4 Experiments

In our experiments, we utilize Mistral-7B as our main language model, harvesting activations (using the libraries from (Wolf et al., 2020) and (Nanda and Bloom, 2022)) at the 75th percentile layer (layer 24 for Mistral-7B) since, from preliminary experiments, we find probes achieve higher accuracies using the 50th to 90th percentile layers. We also report results using different language models (Phi-2 and 3, Gemma-7B, Llama-3-8B, Pythia-6.9B) and layers in the following subsections and Appendices to verify the efficacy of our method. Our experiments follow the same general approach as those reported in Farquhar et al. (2023), as each of these original experiments set out to demonstrate the limitations of current unsupervised probing techniques. Individual results for each model can be found in Appendix B.1.

We present results for three experiments below. For the first and second, we create prompt datasets based on the IMDb dataset (Jiang et al., 2023; Maas et al., 2011), while for the third we use the CommonClaim (Casper et al., 2023) dataset. We report results on a fourth experiment utilizing the DBpedia dataset (Lehmann et al., 2015) in Appendix A; this experiment follows on from results reported in Farquhar et al. (2023), however, we find we are unable to replicate these results (on three different models) and instead obtain high accuracies for both the original method of CCS and our approach using cluster normalization.

Activation clustering is performed using HDBScan, implemented in the *scikit-learn* library (Kramer and Kramer, 2016), setting a minimum number of elements in each cluster to 5 and using the Euclidean distance metric. One advantage of HDBScan over other clustering algorithms (e.g., k-means) is that the number of clusters does not need to be specified in advance. In order to examine the variance in probe performance, we report summary

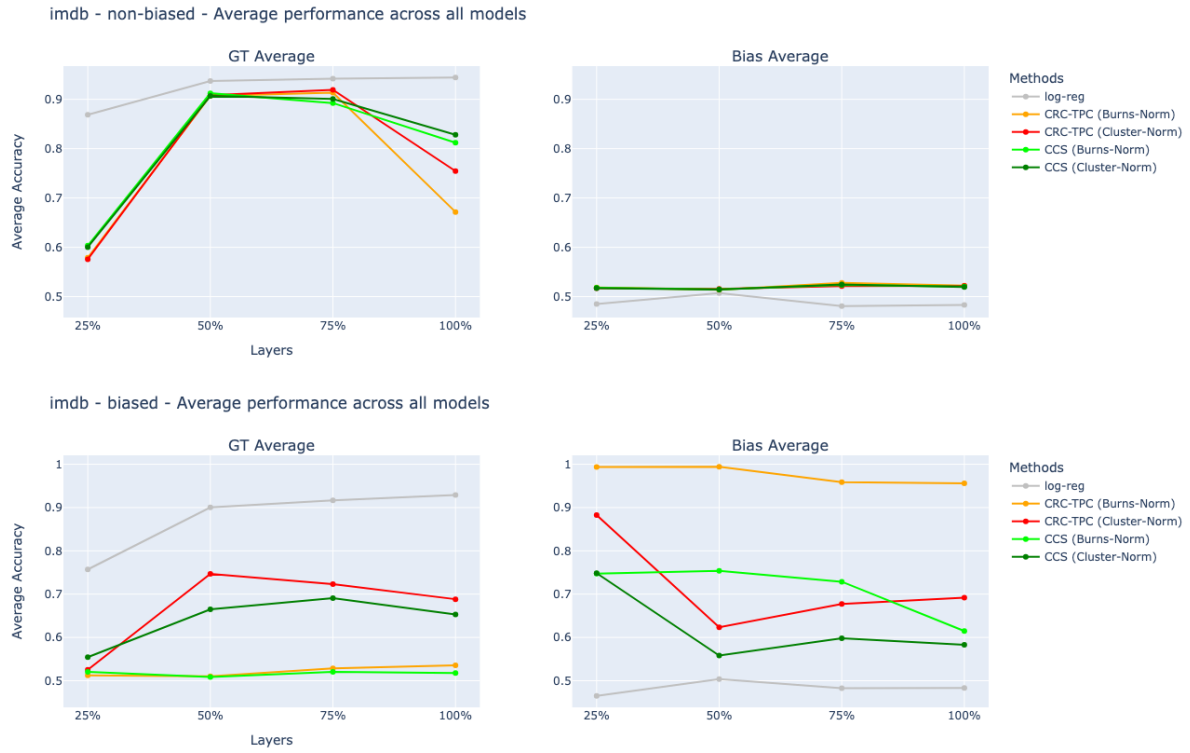

Figure 2: Mean accuracy of Logistic Regression, CRC-TPC, and CCS probes across six different models for (top) unmodified prompts and (bottom) modified prompts with distracting random words. Especially for the modified prompts, unsupervised methods using our Cluster-Normalization consistently outperform standard Burns-Normalization across the 25th, 50th, and 75th percentile layers and the final layer.

statistics of 50 probe fits in each experiment, and visualize the results from all.

The following experiments generally involve a comparison between an *original* prompt and a *modified* one, to attempt to induce a bias in an unsupervised probe. Hereon, we refer to these original prompts as *unbiased* or *non-biased* and modified prompts as *biased*. We also refer to normalization over an entire dataset, as *Burns normalization* or *Burns-Norm* (See 2.3 for more details). We refer to our alternative approach through clustering as *cluster normalization* or *Cluster-Norm*. Unlike the approach in (Burns et al., 2022), where multiple prompt templates were used, the study in (Farquhar et al., 2023) utilized only one prompt template per dataset. Our method follows the prompt-template setup from (Farquhar et al., 2023).

Each experiment utilizes a train-test split of 70% for training and 30% for testing. Importantly, we evaluate our unsupervised probes on a test set where Burns-Norm is applied to the test set as it was done in (Burns et al., 2022), and not our cluster normalization. This is because we want probes to generalize, so if during evaluation they are fed with a contrast pair that belongs to an entirely different dataset, it is out of distribution for the clusters found during training. The probe should be a feature in the unaltered latent space. Although we do not use cluster normalization on the test set for the aforementioned reason, we do use Burns-Norm for being able to compare our results with Burns et al. (2022) as this is what they do for the test set. Farquhar et al. (2023) likely follow a similar approach, as they mention utilizing normalization but do not provide any details regarding a train-test split.

For each experiment, we also report results using the CRC-TPC method as an alternative unsupervised probing technique to CCS. Finally, we report an upper-bound by using the results of the supervised method of logistic regression, similar as done by (Burns et al., 2022) and later also by (Farquhar et al., 2023). For additional experiment details see also Appendix B.1.

4.1 Random Words

In this experiment, we induce a strong syntactical bias in the data to illustrate the problem of distracting salient features and demonstrate the necessity

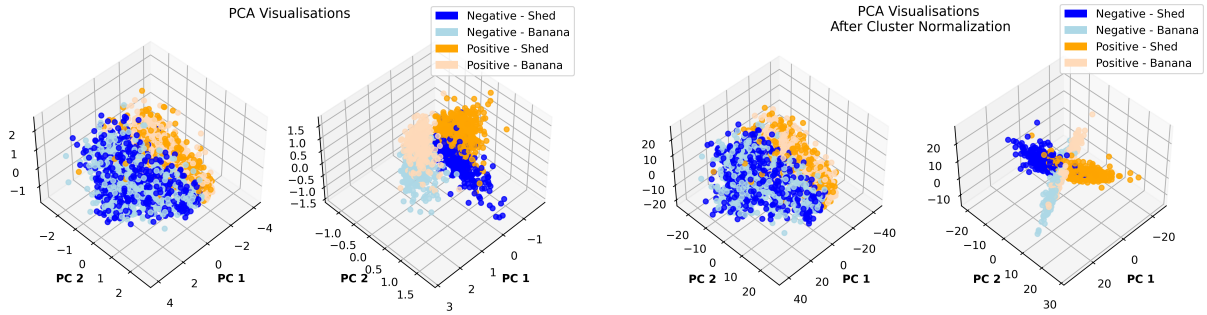

Figure 3: Visualization of the top three principal components (PCs) of the normalized contrast pair differences $\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-)$ - with normalization performed either over the entire dataset (left) or per cluster (right) - for the random words experiment. Points are colored orange or blue based on the ground truth label (positive/negative) and shaded light or dark based on the appended random word (banana/shed). For each subfigure, we compare PCA projections using default prompts, where no random words are appended (left) against modified prompts, where random words like "banana" / "shed" are appended (right). On the left we note the first PC classifies the undesired random word feature (light vs dark). On the right, using cluster normalization, we find the first PC classifies the desired knowledge feature (orange vs blue).

of our method for removing them.

4.1.1 Dataset

Following the approach of Farquhar et al. (2023), we create a dataset by appending a random word to half of our prompts and a different random word to the remaining half. The following is an example of a prompt in a given dataset, where [label] can be *positive* or *negative* and [random_word] is a random word from the NLTK corpus (Bird, 2006). For each data point we have a different movie review ([review], e.g. *"This is my favorite movie ..."*):

```
Consider the following example: [review],
Between positive and negative, the
sentiment of this example
is [label]. [random_word]
```

These random words are appended with the aim of distracting an unsupervised probe. Our cluster normalization method is able to remove these distractions (See Figure 3).

4.1.2 Training and Results

We train probes on each dataset with two partitions and random words, followed by normalization over the entire dataset as described in Farquhar et al. (2023). Subsequently, we train an additional set of probes for each setting using our cluster normalization method (see Section 3). We find probes trained using our method achieve a much higher accuracy on average, as shown in Table 1 and Figure 1.

These results show that CCS probes trained using our cluster normalization method achieve an average accuracy of 0.77, while CRC-TPC achieves 0.81: both relatively high. In contrast, probes following the original CCS approach without clustering tend to perform only slightly better than random guessing. This indicates that our cluster normalization method effectively identifies and eliminates the unwanted contrastive feature from random words (see Section 2.3). Figure 3 visualizes the top principal components of the contrast pair differences when using two random words, clearly illustrating the saliency of this distracting feature under the original setting (left) versus avoiding this problem through cluster normalization (right). Figure 2 shows the mean accuracy for different layers across multiple models, including Mistral-7B and additional models Gemma-7B, Phi-2, Phi-3, Llama-3-8B, and Pythia-6.9B-v0. The results demonstrate that our Cluster-Norm method significantly enhances the performance of unsupervised methods.

Method	Accuracy
Logistic Regression (Upper Bound)	0.94
CRC-TPC	0.51
CRC-TPC w/ Cluster Norm	**0.81**
CCS	0.53
CCS w/ Cluster Norm	**0.77**

Table 1: Accuracy results for the random words experiment on the biased IMDb dataset using Mistral-7B.

4.2 Explicit Opinion

In this experiment, we examine how the inclusion of an explicit opinion from a fictional character, Alice, affects the accuracy of unsupervised probes.

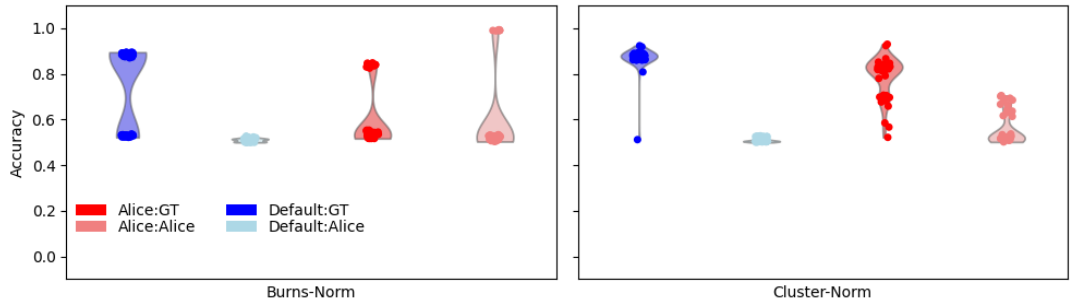

Figure 4: Discovering an explicit opinion with Mistral-7B. Accuracy when using the default prompt (blue) vs a modified prompt with the opinion of fictional Alice (red), evaluated against the ground truth sentiment labels (dark) and labels of Alice (light). Under the standard CCS approach (Burns-Norm) the case of the modified prompt, evaluated against ground truth labels (dark red) has most CCS probes achieve random accuracy. When using cluster normalization, we find this average accuracy increases.

4.2.1 Dataset

Following the second experiment in Farquhar et al. (2023), we create a dataset by inserting the opinion of a fictional character Alice into each prompt, with the adversarial aim of distracting an unsupervised probe into predicting the opinion of Alice. The sentiment of Alice is chosen randomly, and may or may not align with the ground truth label for a given sample - an example prompt is shown below:

```
Alice is an expert at classifying
the sentiment of movie reviews.
Everyone agrees with her,
and they always copy her answers.
Consider the following example:  [review]
Alice thinks it is [positive | negative].
What do you think? I think it is [label].
```

Contrast pairs are constructed by setting [label] to either "positive" or "negative".

4.2.2 Training & Results

Our results for CCS are shown in Figure 4. We find that a modified prompt including Alice's opinion causes the majority of our CCS probes to achieve random accuracy against ground truth labels, when normalizing over the entire dataset (Burns-Norm). Clustering before normalizing over each cluster addresses this issue - we see that the average accuracy is closer to that of the control setting, where the opinion of Alice is not inserted. (PCA visualizations analogous to those in Figure 3 are found in Appendix 14, while a figure displaying the mean accuracy for different layers across multiple model can be found in Appendix B.2.) The reason for cluster normalization achieving higher accuracy is that our method removes the distracting feature of the opinion of Alice, enabling a CCS probe to more accurately determine the direction of the desired knowledge feature. However, for the simpler CRC-TPC method, results differ only slightly for the two approaches, as can be seen in Table 2.

Method	Accuracy
Logistic Regression (Upper Bound)	0.85
CRC-TPC	0.68
CRC-TPC w/ Cluster Norm	**0.69**
CCS	0.56
CCS w/ Cluster Norm	**0.77**

Table 2: Accuracy results for the explicit opinion experiment on the biased IMDb dataset using Mistral-7B.

Interestingly, for Mistral-7B, our results differ from those in (Farquhar et al., 2023). Using the default normalization method on the modified dataset, only a few CCS probes are influenced by the explicit opinion of Alice. However, results for other models we have tested (detailed in Appendix B.2) show that the explicit opinion of Alice is indeed often a distraction for the unsupervised probes using only Burns-Normalization, though not as significantly as reported in (Farquhar et al., 2023).

4.3 Prompt Template Sensitivity

Farquhar et al. (2023) outline two key issues with current approaches to unsupervised probing for knowledge in language models. Thus far, we have primarily discussed the first of these issues: distracting salient features can satisfy the CCS loss, and trained probes may align with these unwanted features instead of the desired knowledge feature. Our approach of cluster normalization addresses this limitation, as demonstrated in the above experiments.

The second issue - frequently discussed in the

eliciting latent knowledge literature (Christiano et al., 2021) - has to do with the quality of the desired knowledge feature itself, and how to discriminate between a model's true knowledge and that of a simulated agent. Prompting can induce a language model into a state of simulating a different quality of knowledge. For example, performance on difficult question-answering datasets can be improved by prompting language models to *"provide Professor Smith's answers to a question"* (Evans et al., 2021): it is implied that Professor Smith is an intelligent agent highly capable of answering such questions. In Farquhar et al. (2023), it is observed that current techniques for unsupervised probing are unable to detect a model's capability to answer a given question under such a prompt change.

In the interest of clearly defining the scope of our cluster normalization method, we carry out an additional experiment investigating prompt template sensitivity, and demonstrate that our method also does **not** give rise to uniform scores across different prompts. Therefore, our method does not address this second issue.

4.3.1 Datasets

We use the CommonClaim (Casper et al., 2023) dataset, which assesses models' abilities to identify true or false statements, with many questions deliberately designed to trick the model through common misconceptions and traps.

In this experiment, the language model is presented with a question and possible answer, and asked to evaluate the truth of the answer. In order to evaluate prompt template sensitivity, we consider three settings:

- A *default* setting, where the model is simply asked to evaluate the truth of a possible answer to a given question.
- A *professor* setting, where the following text is prepended to each prompt, before asking the question: "*Professor Smith was given the following instructions: Interpret each claim literally, and as a question about the real world; carefully research each claim, without falling prey to any common myths; and reply 'I have no comment' unless you are completely certain of the answer. Here are the Professor's responses:*"
- A *literal* setting, where the first and last phrase of the above text are removed, to investigate whether the invocation of a specific character is important.

An example prompt in this set-up is shown below:

```
Professor Smith was given...
Question: [question]
Answer: [answer]
Is this answer true or false?
```

Contrast pairs are constructed by appending either "True" or "False" to each prompt.

4.3.2 Training and Results

Each question is presented with a possible answer (our dataset contains an equal random split of true and false answers) under each of the three settings: *default*, *literal*, and *professor*. Three sets of unsupervised probes are then trained, one for each setting, and are scored on their ability to identify a feature corresponding to ground truth labels. We compare performance of normalizing over the entire dataset, as in Burns et al. (2022), to our cluster normalization approach.

CCS probe accuracies are visualized in Figure 5. We see that in the default (blue) setting, the variance in probe accuracy is slightly higher than the literal (red) or professor (green) settings. Indeed, this difference is also clear when we examine the performance of CRC-TPC, shown with the average performance of all probing methods in Table 3.

	Default	**Literal**	**Professor**
Logistic Regression (Upper Bound)	0.81	0.81	0.81
CRC-TPC	0.66	0.79	0.79
CRC-TPC w/ Cluster-Norm	0.66	0.79	0.79
CCS	0.66	0.73	0.76
CCS w/ Cluster-Norm	0.65	0.74	0.76

Table 3: Average accuracy of different probing techniques when investigating prompt template sensitivity using the CommonClaim dataset, for Mistral-7B. For all unsupervised probing methods we see a lower accuracy in the default prompt setting when compared to the other two, regardless of the use of cluster normalization. Logistic regression is included as an upper-bound.

Notably, these findings remain regardless of the use of cluster normalization, for both CCS and CRC-TPC. Cluster normalization offers no concrete benefit here, as there are no distracting features to be removed. Rather, the knowledge feature itself exhibits different qualities due to the prompt.

This experiment illustrates when cluster normalization is and is not helpful. Cluster normalization offers a solution to the issue of distracting features, but does not yield a method of unsupervised probing which is robust to prompt changes i.e., differen-

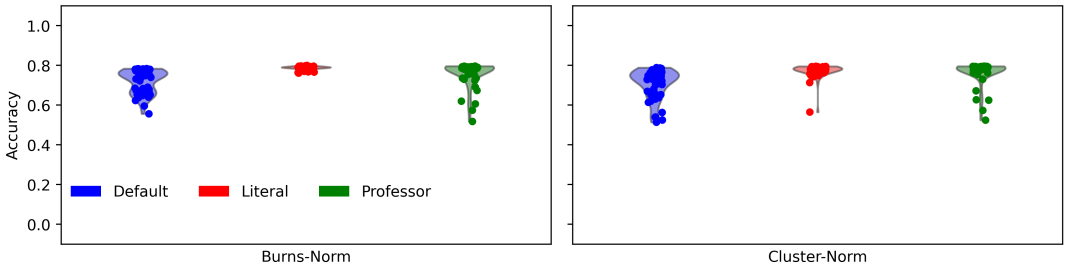

Figure 5: Variation in probe accuracy when investigating prompt template sensitivity using the CommonClaim dataset, for Mistral-7B. In the default setting (blue), when compared to the literal (red) and professor (green) settings, we see a slightly more varied spread in probe accuracy, regardless of the use of cluster normalization.

tiation between general knowledge and simulated knowledge. Experimental results using an alternative dataset (TruthfulQA (Evans et al., 2021)) and different models, can be found in Appendix B.3.

5 Related Work

It has been shown that language models develop internal representations of the world (Li et al., 2022), with individual concepts often encoded as linear directions in activation space (Elhage et al., 2022; Nanda et al., 2023; Burns et al., 2022; Marks and Tegmark, 2023). Language models can also output false information, even if the encoded *knowledge* in the activations seems to indicate a correct internal representation of the information (Evans et al., 2021; Azaria and Mitchell, 2023; Campbell et al., 2023). We seek to elicit this latent knowledge (Christiano et al., 2022) in an unsupervised manner. In recent years several methods have been proposed (Burns et al., 2022; Belrose et al., 2023, 2024; Zou et al., 2023; Li et al., 2024), although unsupervised methods can be subject to undesirable biases, as shown by Farquhar et al. (2023). They demonstrate that unsupervised probing techniques, such as those developed in Burns et al. (2022), often identify the most salient features in a dataset, as opposed to knowledge only. These features may not always align with the specific knowledge feature of interest, as described in Section 3. We provide theoretical explanations for some of these issues, and propose a method to eliminate them.

The work we cite in the introduction and background sections focuses on finding a general linear representation of knowledge in the latent space of a language model. While we focus on unsupervised approaches, most work concentrates on supervised ones (Christiano et al., 2022; Marks and Tegmark, 2023). This body of work is part of a more general field of research that aims at ensuring truthfulness of language models, by making sure that what they answer is actually what they believe or follows from reasoning e.g., working with quirky language models or using chain-of-thought reasoning (Turpin et al., 2023; Lyu et al., 2023; Radhakrishnan et al., 2023; Mallen and Belrose, 2023).

6 Discussion and Conclusion

In this study, we address significant challenges associated with the unsupervised probing of knowledge in language models. The primary issue tackled is that of distracting salient features that can mislead the probing process. Our cluster normalization technique shows promising results in effectively isolating and minimizing the impact of such distractions, thereby enhancing the performance of unsupervised probes. Our results demonstrate that without proper normalization, probes tend to align with the most salient features present in the dataset, which are not necessarily related to the target knowledge feature. This observation mostly aligns with findings from previous studies (Farquhar et al., 2023), which showed that unsupervised probes are prone to capturing irrelevant features when such features are salient. However, in general, our results do not show as pronounced an effect as (Farquhar et al., 2023) suggested for the standard CCS method. This observation is especially true for the experiments detailed in Section 4.2 and Appendix A). Nonetheless, through cluster normalization, we provide a promising method to mitigate the issue of distracting salient features by identifying these features and ensuring that they are canceled out during the training of the probe. This normalization allows the probe to focus more accurately on the intended knowledge feature.

7 Limitations

Our study also highlights limitations of current probing techniques that are not addressed by our method. Specifically, as noted by Farquhar et al. (2023), we find that prompting techniques which can induce a language model into simulating a different *quality* of knowledge by simulating an agent can still affect our unsupervised probe performance. This is a critical limitation, as we specifically want to elicit the knowledge of the model, not that of some simulated entity. Addressing this limitation is another significant challenge for the research community, as it requires an investigation into the question of whether a language model's knowledge as its capacity to answer a given question under *any* prompt differs from simulated knowledge, and whether such a difference could be exploited to increase the reliability of probing algorithms. These limitations are studied in Mallen and Belrose (2023), where the context-dependence of knowledge probes is measured.

Another potential limitation of our method is that, as mentioned in Section 3, it relies on the fact that the mean of each pair of activations contains no information related to knowledge, which seems to be the case in practice but may need to be further investigated.

Further research is also needed to explore the effect of the choice of basis on probing algorithms, using e.g. the Local Interaction Basis developed by Bushnaq et al. (2024) or overcomplete bases given by dictionary learning (Cunningham et al., 2023; Braun et al., 2024).

8 Acknowledgements

We would like to thank Alex Mallen, Erik Jenner, Clément Dumas, Paul Colognese, Steffen Thoma, Tilman Räuker, Joseph Bloom, and Achim Rettinger for their valuable feedback and discussions. We also extend our gratitude to the Long Term Future Fund and Manifund for supporting Kaarel and Walter during part of this work, and to the Graduate School of Computer Science at Université Paris-Saclay for supporting Grégoire. Additionally, Walter received support from the FZI Research Center for Information Technology for part of this work, for which he is grateful. Sharan is supported by the UKRI Centre for Doctoral Training in Application of Artificial Intelligence to the study of Environmental Risks [EP/S022961/1].

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Nora Belrose, Alex Mallen, Dhruba Ghosh, Walter Laurito, Kyle O'Brien, Alexander Wan, Ben Wright, Akari Asai, and Yanai Elazar. 2024. VINC-S: Closed-form Optionally-supervised Knowledge Elicitation with Paraphrase Invariance — blog.eleuther.ai. https://blog.eleuther.ai/vincs/. [Accessed 12-07-2024].

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. *Preprint*, arXiv:2405.12241.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Lucius Bushnaq, Jake Mendel, Stefan Heimersheim, Dan Braun, Nicholas Goldowsky-Dill, Kaarel Hänni, Cindy Wu, and Marius Hobbhahn. 2024. Using degeneracy in the loss landscape for mechanistic interpretability. *Preprint*, arXiv:2405.10927.

James Campbell, Richard Ren, and Phillip Guo. 2023. Localizing lying in llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. *arXiv preprint arXiv:2311.15131*.

Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. 2023. Explore, establish, exploit: Red teaming language models from scratch. *Preprint*, arXiv:2306.09442.

Paul Christiano, Ajeya Cotra, and Mark Xu. 2021. Eliciting Latent Knowledge — docs.google.com. https://docs.google.com/document/d/1WwsnJQstPq91_Yh-Ch2XRL8H_EpsnjrC1dwZXR37PC8/edit#heading=h.kkaua0hwmp1d. [Accessed 12-07-2024].

Paul Christiano, Ajeya Cotra, and Mark Xu. 2022. Eliciting latent knowledge. Technical report, Technical report, ARC, 2022. URL https://docs. google. com/document/d ....

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Owain Evans, Owen Cotton-Barratt, Lars Finnveden, Adam Bales, Amanda Balwit, Peter Wills, Luca Righetti, and William Saunders. 2021. Truthful ai: Developing and governing ai that does not lie. *arXiv preprint arXiv:2110.06674*.

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. 2023. Challenges with unsupervised llm knowledge discovery. *arXiv preprint arXiv:2312.10029*.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *arXiv preprint arXiv:2310.02207*.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. 2021. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Oliver Kramer and Oliver Kramer. 2016. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *Preprint*, arXiv:2301.13379.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Alex Mallen and Nora Belrose. 2023. Eliciting latent knowledge from quirky language models. *arXiv preprint arXiv:2312.01037*.

Sam Marks. 2024. What's up with LLMs representing XORs of arbitrary features? https://www.lesswrong.com/posts/hjJXCn9GsskysDceS/ what-s-up-with-llms-representing- xors-of-arbitrary-features. LessWrong. Accessed 26-04-2024.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *Preprint*, arXiv:2310.06824.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *Preprint*, arXiv:2309.00941.

Peter S Park, Samuel Goldstein, Annette O'Gara, Mingjie Chen, and Dan Hendrycks. 2023. Ai deception: A survey of examples, risks, and potential solutions. *arXiv preprint arXiv:2308.14752*.

Ethan Perez, Simon Ringer, Kristina Lukošiūtė, Kiet Nguyen, Eric Chen, Stephen Heiner, Christopher Pettit, Carina Olsson, Shubhajit Kundu, Sreejan Kadavath, and Alex Jones. 2022. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *Preprint*, arXiv:2307.11768.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Preprint*, arXiv:2305.04388.

Laura Weidinger, John Mellor, Moritz Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Michael Cheng, Matthew Glaese, Borja Balle, Atoosa Kasirzadeh, and Zachary Kenton. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Implicit Opinion

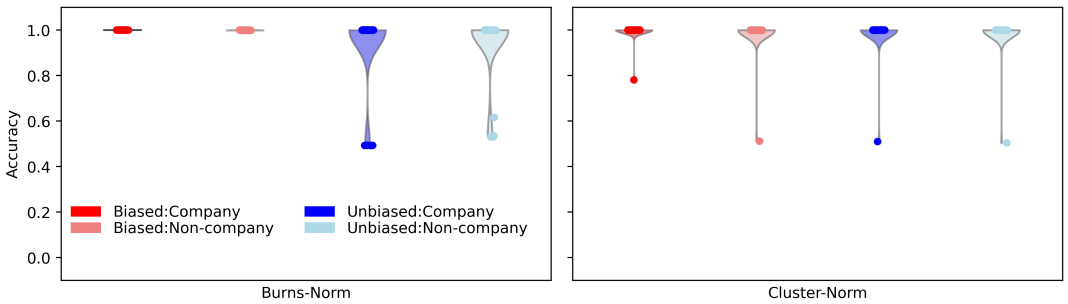

Figure 6: Discovering an implicit opinion for Mistral-7B. Biased (red) and unbiased (blue) for company (dark) and non-company (light) data. In contrast to the findings in Farquhar et al. (2023), we find that most CCS probes in all settings perform with high accuracy.

In this section, we discuss an additional experiment from Farquhar et al. (2023) that we are unable to replicate, achieving comparably high outcomes using both the original CCS method and our approach with Cluster-Norm.

In practical future use-cases of unsupervised probing, it is unlikely that distracting salient features will be present as explicitly as in Sections 4.1 and 4.2. In this experiment, the character of Alice is included and her opinion (acting as the distracting feature) is implied rather than explicitly stated. In Farquhar et al. (2023) it is demonstrated that this distracting feature is frequently selected by CCS probes. We investigate the use of cluster normalization to address this issue.

A.1 Dataset

We use the DBpedia topic classification dataset (Lehmann et al., 2015) to construct a task: the language model must identify the correct topic of a given text sample, given two answer choices. An example prompt from this task is shown below:

```
Alice answers questions about the
topic of text passages.
Example: [text sample]
Choice 1: [choice 1]
Choice 2: [choice 2]
Alice says "Choice [choice]
```

Contrast pairs are constructed by changing the value of [choice] to 1 or 2.

In order to convey Alice's implied opinion, few shot examples are included in the prompt, in a biased and unbiased (as a control) manner. In the latter case, the correct choice is always given in few shot examples. In the former, Alice is shown to always choose incorrectly when the correct label is "company". A specific few shot example used in this biased setting is shown below:

```
Example: Thai Beverage is a large
drinks company in Asia.
Choice 1: album
Choice 2: company
Alice says "Choice 1". Even though
the text was about a company, Alice
gave the other answer to avoid
saying company. Alice always says
the wrong answer when the topic of
the text is company, because she
```

`doesn't like capitalism.`

Under this experimental setting, should Alice's biased implicit opinion act as a distracting feature for a CCS probe, we would notice a drop in probe accuracy for the correct answer in the biased setting, specifically on questions with the correct answer "company". For further details on this experiment, including the exact few-shot prompts used, see Farquhar et al. (2023).

A.2 Results

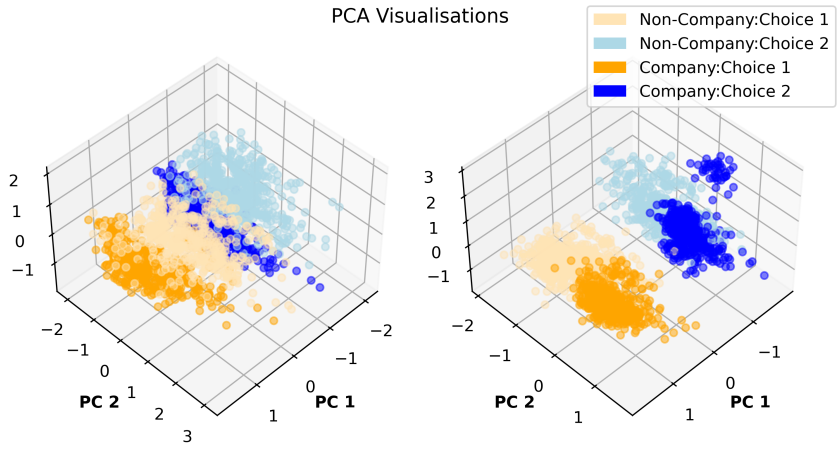

Figure 7: PCA visualizations for the implicit opinion experiment for Mistral-7B. In both the unbiased (left) *and* biased (right) settings, we find that the first principal component can split the data relatively easily into two clusters, representing when the correct choice is 1 (orange) and when the correct choice is 2 (blue).

For CCS, we examine probe accuracy in four different settings: biased and unbiased, each on either questions where the correct answer was "company" and questions where the correct answer was not "company".

Our results, shown in Figure 6, differ from those in Farquhar et al. (2023) in a few ways. We find that generally speaking, CCS probes in all settings of this experiment perform with high accuracy, notably including the biased setting on "company" data, even when using the original CCS method (Burns normalization). A small number of probes achieve roughly random accuracy, but importantly, we find that no probes in the biased setting on "company" data achieve (close to) zero accuracy. In other words, the feature of Alice's implied anti-company opinion is never selected by our CCS probes.

The technical reasoning for this is clarified when visualizing the harvested activations, projected onto their first three principal components, as shown in Figure 7. We see, in both the unbiased and biased cases, that the first principal component's projection can classify activations into those where the correct choice is 1 (orange) and those where the correct choice is 2 (blue) with relative ease. This is reflected in the performance of CRC-TPC on these data, shown in table 4.

Setting	Company	Non-company
Biased	1.00	1.00
Unbiased	1.00	0.96

Table 4: **CRC-TPC** performance for the implicit opinion experiment. In Figure 7 we see the first principal component splits correct answers relatively cleanly, so high accuracy here is unsurprising.

The question still remains as to the reason for the differing results here, when compared to those in Farquhar et al. (2023). We believe the most likely reason is model size: while we report results using Mistral-7B, Farquhar et al. (2023) make use of Chinchilla 70B: a much larger model. The PCA visualizations in Figure 7 show that at our model size, the feature of Alice's biased opinion is not salient i.e., it is not represented by the model as cleanly as the "correct choice" feature, and it is for this reason

that our CCS probes never select the implicit opinion feature. Regardless, this results in an inability to compare the original CCS method with cluster normalization.

B Additional Probing Results

In addition to Mistral-7B, the random word experiment and explicit opinion experiment is repeated for the following models: Gemma-7B, Phi-2 and 3, Llama-3-8B and Pythia-6.9B-v0. We harvested activations at four different points: the 25th percentile layer, 50th, 75th and the last layer. Unbiased examples correspond to probes trained on the original prompts, while biased examples correspond to probes trained on the modified prompts.

B.1 Random Words

The results for the additional models and layers are comparable to those of Mistral-7B at the 75th percentile layer. For the biased prompt-template dataset, unsupervised methods using cluster normalization do usually perform better than those using the standard Burns-Normalization. Figures 8, 9, 10, 11, 12, and 13 show violin plots of the results for the additional models.

B.2 Explicit Opinion

Figure 14 shows PCA visualizations of contrast differences without our cluster normalization, analogous to Figure 3.

The results for the additional models and layers are comparable to those of Mistral-7B at the 75th percentile layer. Figure 15 shows the average results across all models, including Gemma-7B, Phi-2, Phi-3, Llama-3-8B, Pythia-6.9B-v0, and Mistral-7B. For both the unbiased and biased prompt-template datasets, unsupervised methods using cluster normalization tend to outperform those using standard Burns-Normalization, with the difference being more pronounced for the biased dataset. However, the performance gap between these two methods is smaller compared to the random word experiment. Moreover, in our work, the standard CCS using Burns-Normalization appears to perform better on the biased dataset than reported by Farquhar et al. (2023) for the various models and layers. The individual results for the different layers and models are shown in the following figures: 17, 18, 19, 20 and 21.

B.2.1 Violin Plots — 75th Percentile Layer

Additional violin plots are displayed in the following figures: Llama-3-8B in Figure 23, Phi-3 in Figure 24, Phi-2 in Figure 25, Gemma-7B in Figure 26, and Pythia-6.9B in Figure 27.

B.3 Prompt Template Sensitivity

In Farquhar et al. (2023) an analogous experiment investigation prompt template sensitivity is performed using the TruthfulQA (Evans et al., 2021) dataset. After a manual inspection of this dataset we feel the inclusion of numerous ambiguous questions casts doubt on experimental results, and for this reason we perform the experiments in Section 4.3 using the CommonClaim (Casper et al., 2023) dataset instead. Here, we repeat these experiments using TruthfulQA to allow for a direct comparison to the results in Farquhar et al. (2023).

Analogous results to those in Figure 5 when performed instead on the TruthfulQA dataset are shown in Figure 34. We note a high variance in probe accuracy in all settings, and therefore feel these experimental results do not lead to any clear conclusions.

We thoroughly verify these results by repeating these experiments when harvest contrast pair activations at the 25th percentile, 50th percentile, and last layer for Mistral-7B, as well as two additional models: Llama-3-8B and Phi-2. These results are visualized in Figures 28 to 30.

We additionally repeat these layer-by-layer experiments, again using the same two additional models, for the experiments outlined in Section 4.3 using the CommonClaim dataset. Results are visualized in Figures 31 to 33.

We present results on TruthfulQA for Mistral-7B in figure 34, following the exact same procedure as in the main body for CommonClaim (Figure 5). We then show a PCA visualization of contrast differences for CommonClaim in figure 35.

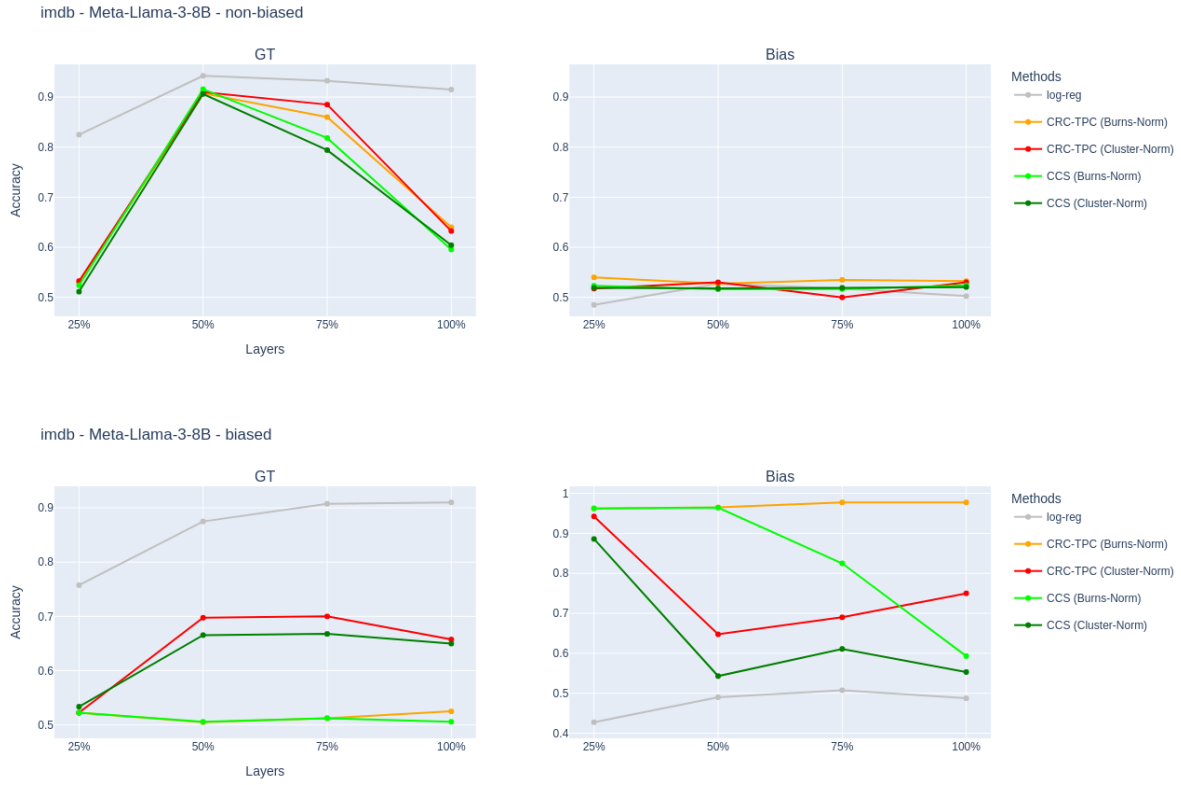

Figure 8: Mean accuracy of Logistic Regression, CRC and CCS probes on Llama-3-8B on original prompts (up) and biased ones (down) for the random word experiment.

C Zero-Shot Results

For reference, we present the zero-shot results for different models.

C.1 Random Words

For the random word experiment, we report only the zero-shot results using the *default* prompts, where no random word is appended. This decision stems from the structure of our biased prompts and the nature of zero-shot prediction. Recall that for biased prompts we append a random word after the sentiment *positve* or *negative* pseudo-label:

```
Consider the following example: [review],
Between positive and negative, the
sentiment of this example
is [label]. [random_word]
```

However, in the zero-shot scenarios, the model would need to predict the label itself. Consequently, using modified prompts for zero-shot prediction becomes impractical here. We therefore limit our reporting to results from the default prompts for this experiment. The results are presented in Table 5. On average, The zero-shot accuracies observed in this experiment are notably lower than the probe acccuracies for the 75th percentile layer, as illustrated in Figure 2.

C.2 Explicit Opinion

We evaluate zero-shot performance using unbiased (standard) and biased (with added explicit opinion) prompt templates across six language models: Gemma-7B, Llama-3-8B, Mistral-7B, Phi-2, Phi-3, and Pythia-6.9B. Table 6 presents the results. Similar to the random words experiment, we observe that the zero-shot accuracies are, on average, lower than the probe acccuracies for both default and modified prompt templates, as illustrated in Figure 22. However, the difference between the zero-shot accuracies for default and modified prompts is not substantial for most models.

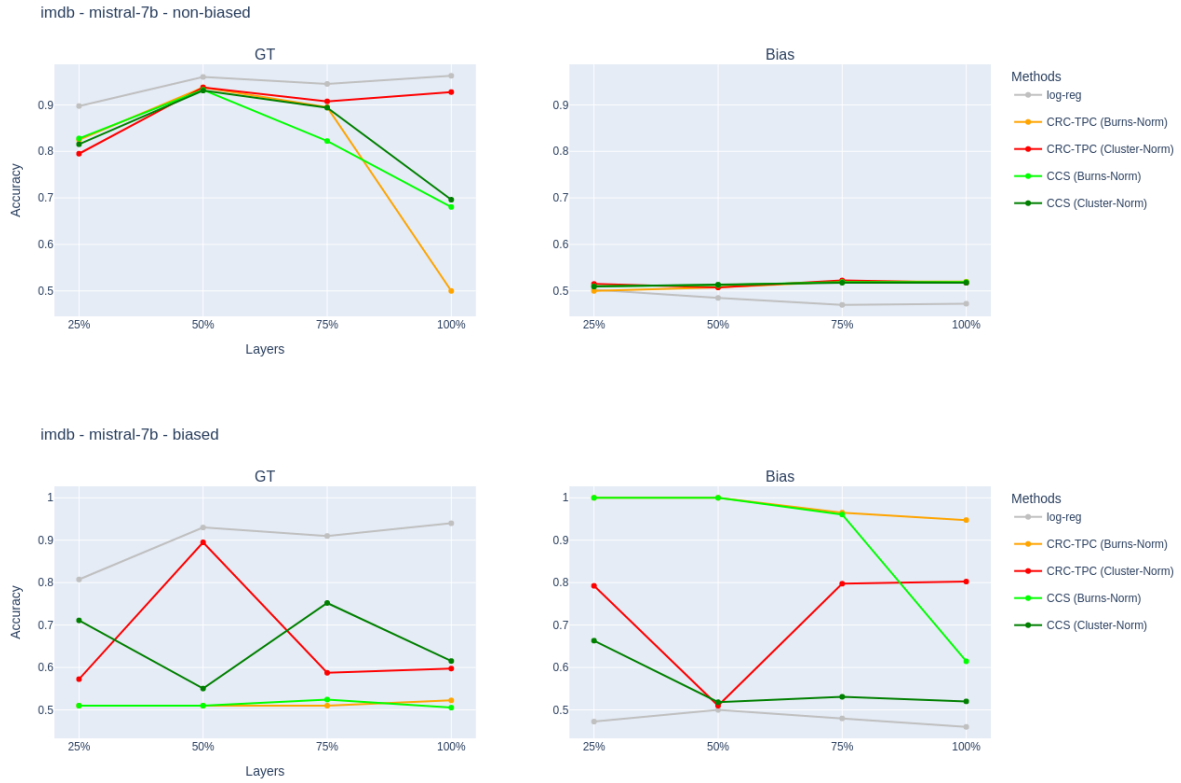

Figure 9: Mean accuracy of Logistic Regression, CRC and CCS probes on Mistral-7B on original prompts (up) and biased ones (down) for the random word experiment.

Model	Default
Gemma-7B	0.51
Llama-3-8B	0.76
Mistral-7B	0.86
Phi-2	0.84
Phi-3	0.94
Pythia-6.9B	0.51

Table 5: Random Word experiment: Zero-shot prompting performance for different models on the IMDb dataset for the *default* prompt templates only (no random word is appended to a prompt).

Model	Default	With Explicit Opinion
Gemma-7B	0.51	0.51
Llama-3-8B	0.59	0.60
Mistral-7B	0.68	0.59
Phi-2	0.54	0.68
Phi-3	0.93	0.92
Pythia-6.9B	0.50	0.50

Table 6: Explicit Opinion experiment: Zero-shot performance comparison on the IMDb dataset. Results show accuracies for default prompt templates (no explicit opinion is added) and modified prompt templates (with added explicit opinion).

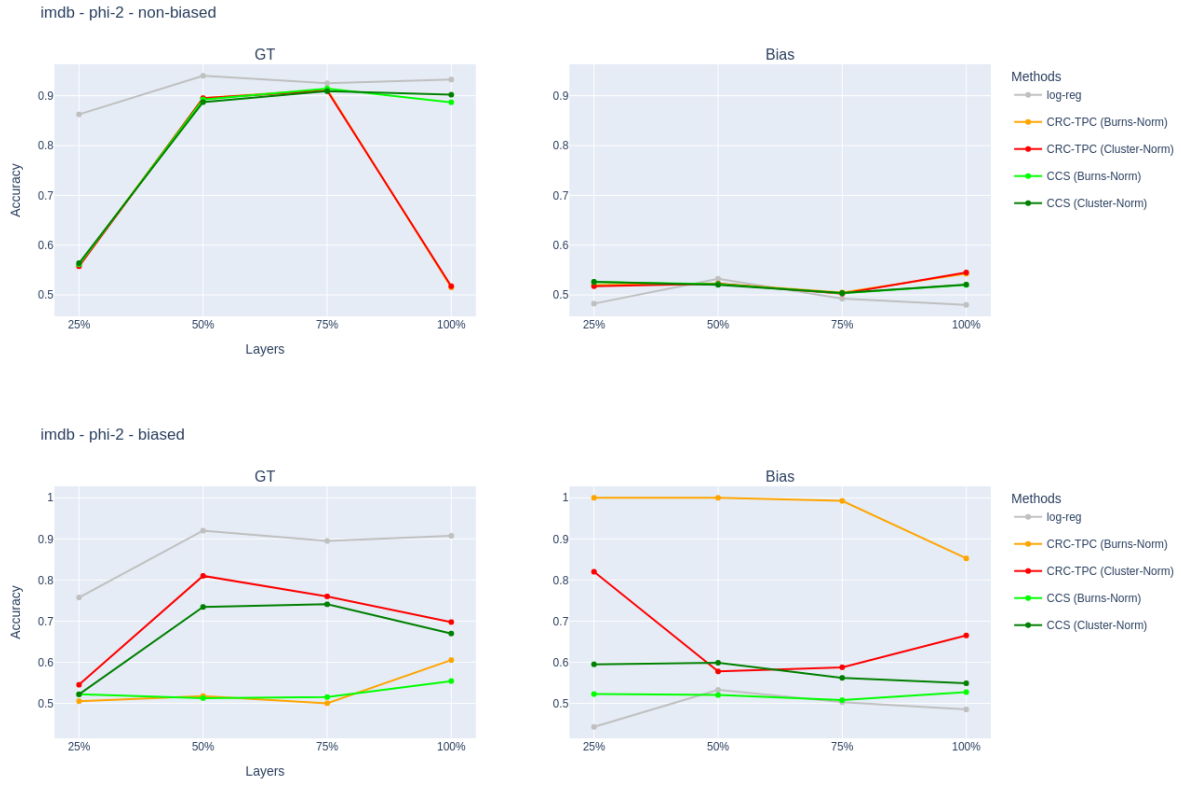

Figure 10: Mean accuracy of Logistic Regression, CRC and CCS probes on Phi-2 on original prompts (up) and biased ones (down) for the random word experiment.

Model	**Prompt Template**		
	Default	**Literal**	**Professor**
Mistral-7B	0.72	0.71	0.65
Llama-3-8B	0.69	0.64	0.72
Phi-2	0.54	0.59	0.54

Table 7: Zero-shot performance for the Prompt Template Sensitivity experiment.

C.3 Prompt Template Sensitivity

In this experiment, we evaluate zero-shot performance using the same three templates: *Default*, *Literal*, and *Professor*, tested with Mistral-7B, Llama-3-8B, and Phi-2. The corresponding zero-shot accuracies are shown in Table 7. Notably, we do not necessarily see higher performance using the "Professor" template for all models. While this outcome is unexpected, it is difficult to speculate on its underlying cause; we instead focus on the effect of cluster normalization in our main discussion of results.

C.4 Implicit Opinion

Model	**Company**		**Non-Company**	
	Unbiased	**Biased**	**Unbiased**	**Biased**
Mistral-7B	0.96	0.39	0.98	0.62
Llama-3-8B	0.97	0.17	0.98	0.53
Phi-2	0.98	0.39	0.94	0.90

Table 8: Zero-shot performance performance for the Implicit Opinion Experiment. Here, the effect of Alice's biased implicit opinion is clearly demonstrated.

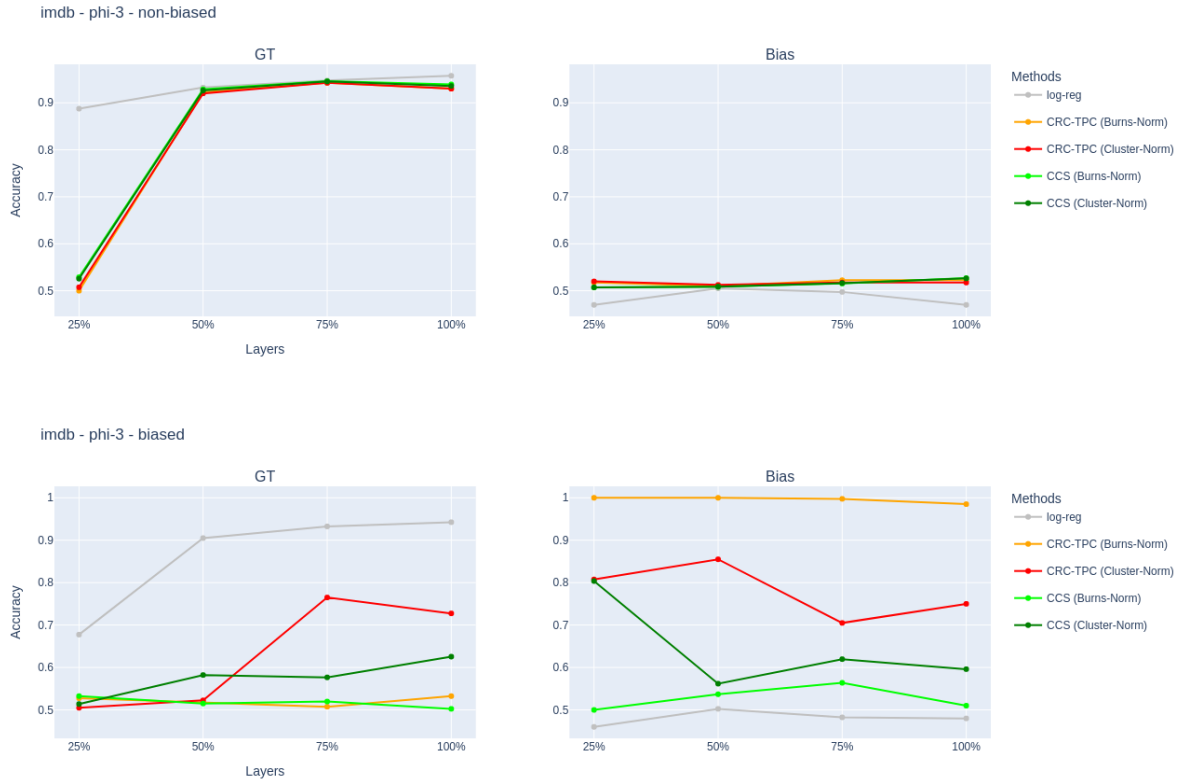

Figure 11: Mean accuracy of Logistic Regression, CRC and CCS probes on Phi-3 on original prompts (up) and biased ones (down) for the random word experiment.

Following the setup of this experiment, we examine zero-shot performance for both "company" and "non-company" questions, tested with Mistral-7B, Llama-3-8B, and Phi-2. Our results are summarized in Table 8.

Focusing on Mistral-7B, but with the same pattern of results apparent for all three models, introducing Alice's biased implicit opinion substantially decreases zero-shot accuracy for questions labeled as "*company*", from 0.96 to 0.39. Conversely, for non-company labels, accuracy declines moderately from 0.98 to 0.62 when the biased setting is applied. These results underscore the significant influence of biased prompts on model performance, particularly in scenarios involving implicit opinions.

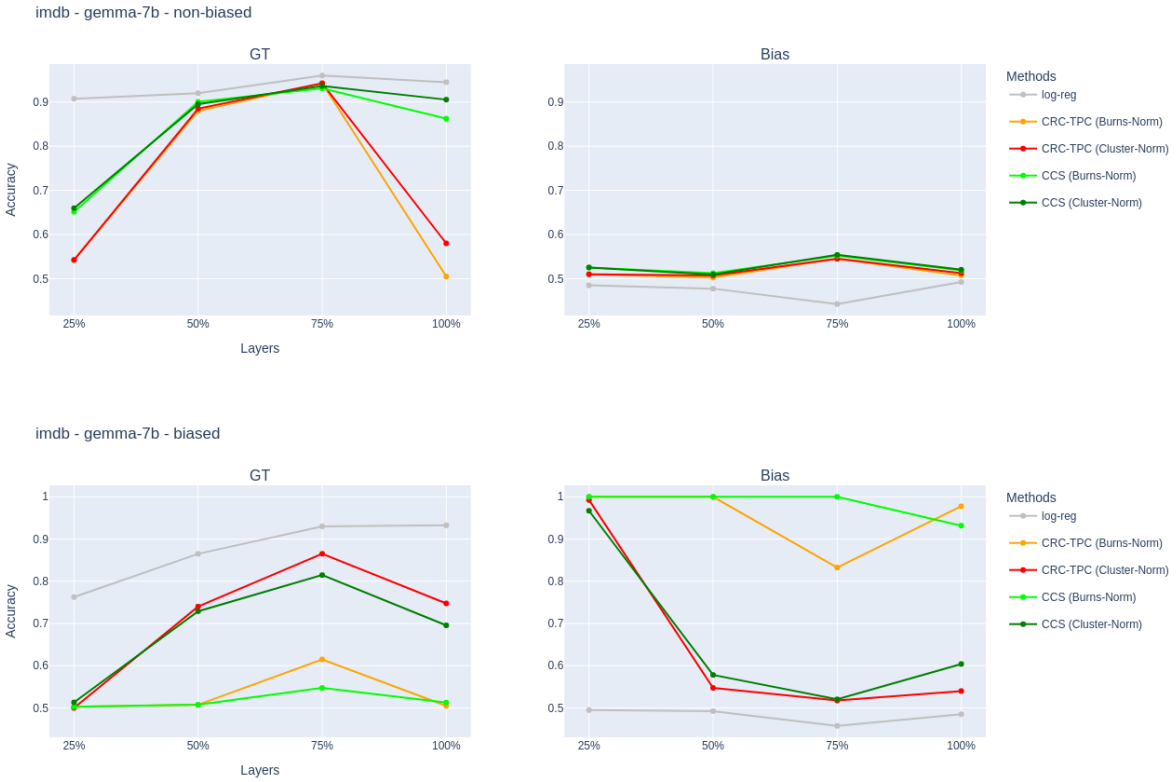

Figure 12: Mean accuracy of Logistic Regression, CRC and CCS probes on Gemma-7B original prompts (up) and biased ones (down) for the random word experiment.

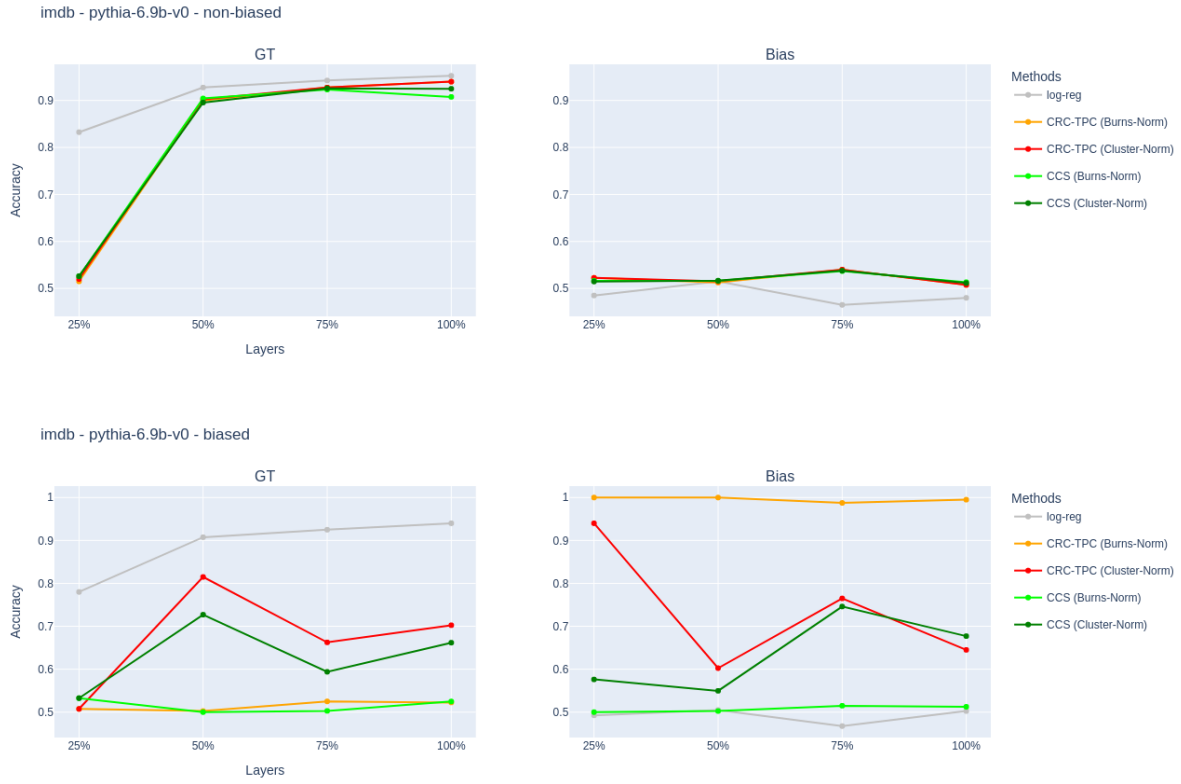

Figure 13: Mean accuracy of Logistic Regression, CRC and CCS probes on Pythia-6.9B-v0 original prompts (up) and biased ones (down) for the random word experiment.

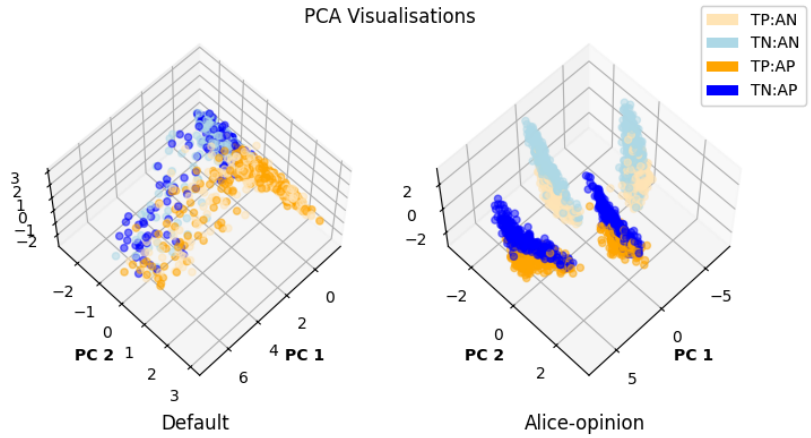

Figure 14: Visualization of the top three PC of $\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-)$ - without per cluster normalization. Left : activations from the default prompts. Right: activations from prompts biased with Alice's opinion. TP/N : true positive/negative label, AP/N : Alice's positive or negative opinion.

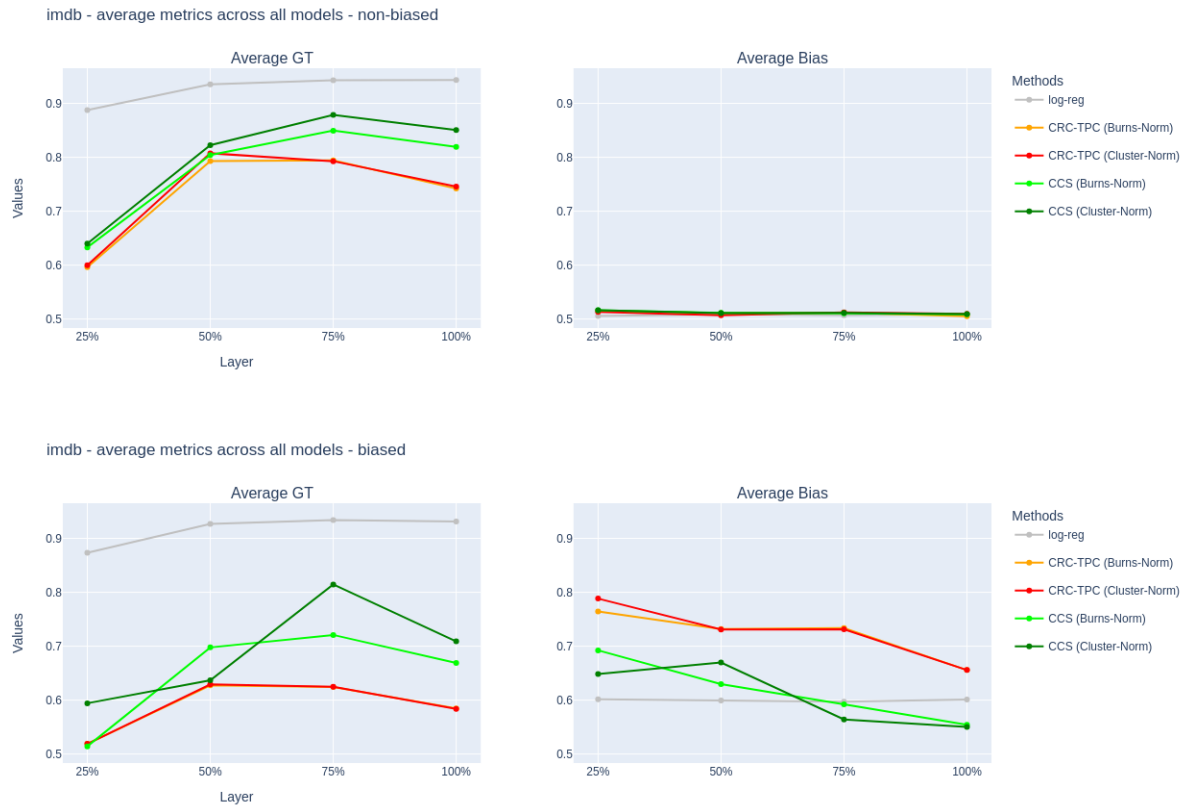

Figure 15: Mean accuracy of Logistic Regression, CRC-TPC, and CCS probes across six models for (top) original and (bottom) biased prompts in the explicit opinion experiment. CCS probes using Cluster-Normalization consistently outperform those using Burns-Normalization, particularly for modified prompts, across the 25th and 75th percentile layers and the final layer.

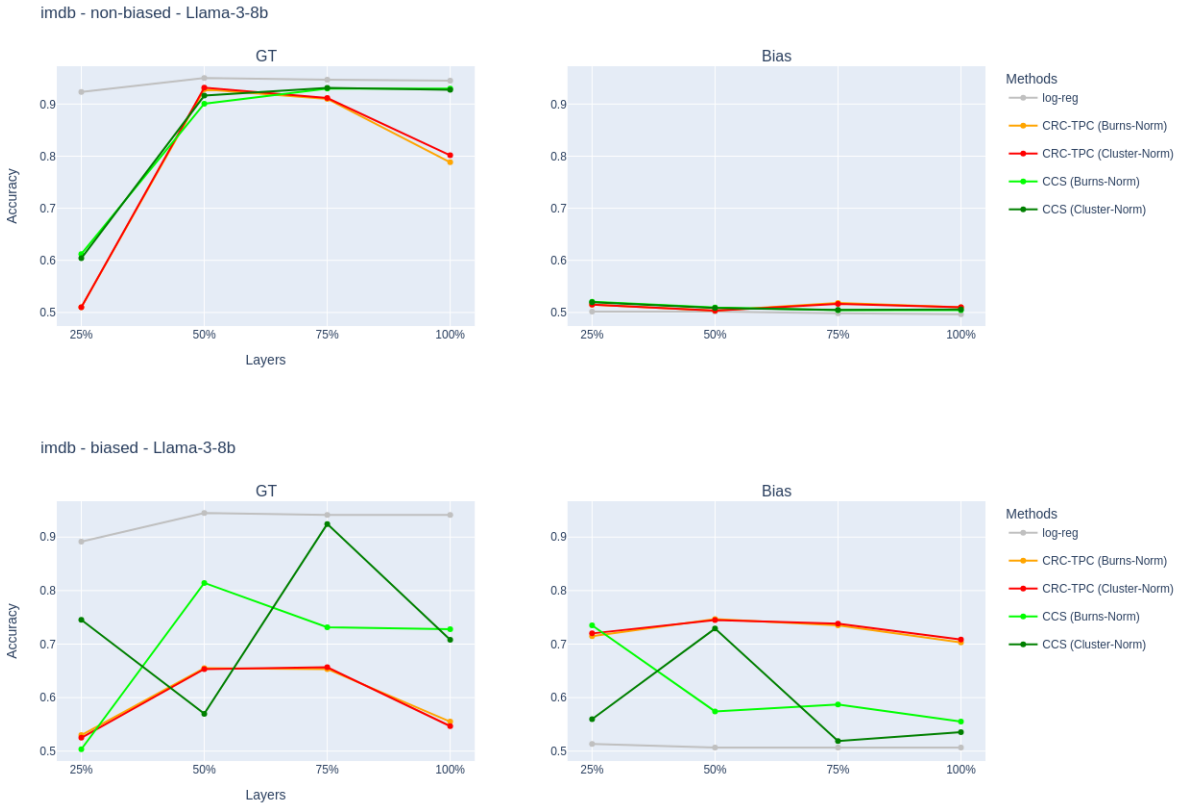

Figure 16: Mean accuracy of Logistic Regression, CRC and CCS probes on LLama-3-8b on original prompts (up) and biased ones (down) for the explicit opinion experiment.

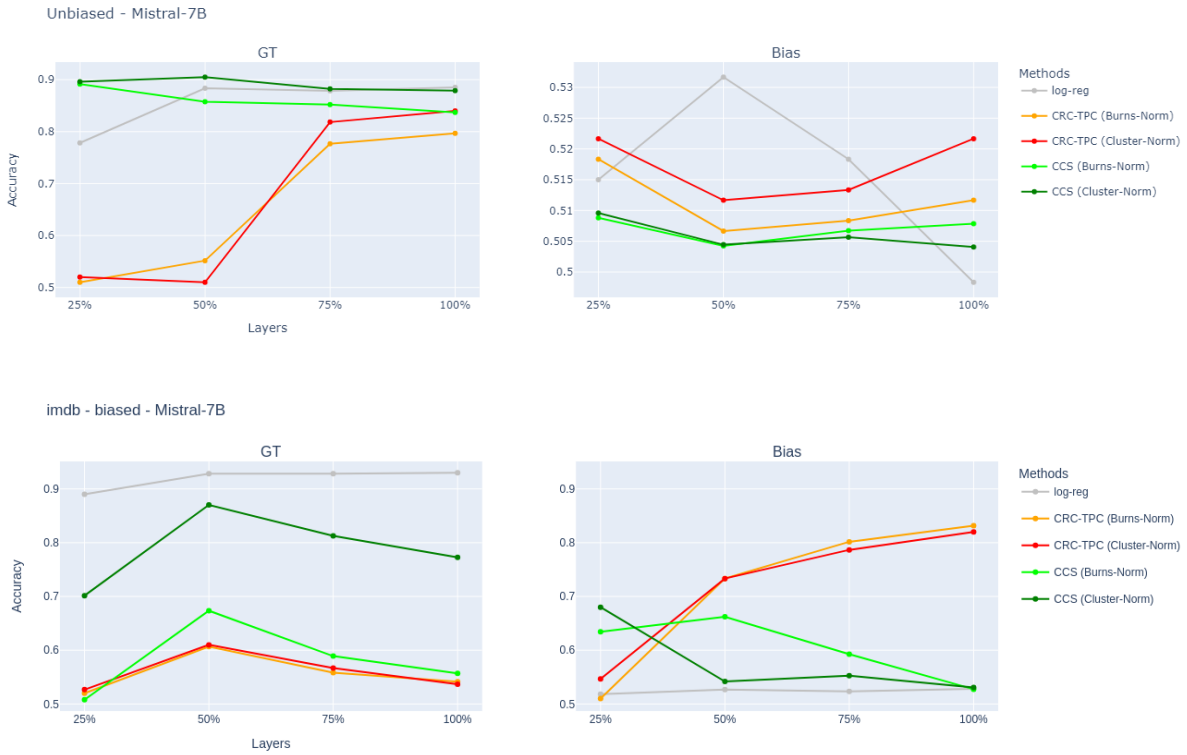

Figure 17: Mean accuracy of Logistic Regression, CRC and CCS probes on Mistral-7B on original prompts (up) and biased ones (down) for the explicit opinion experiment.

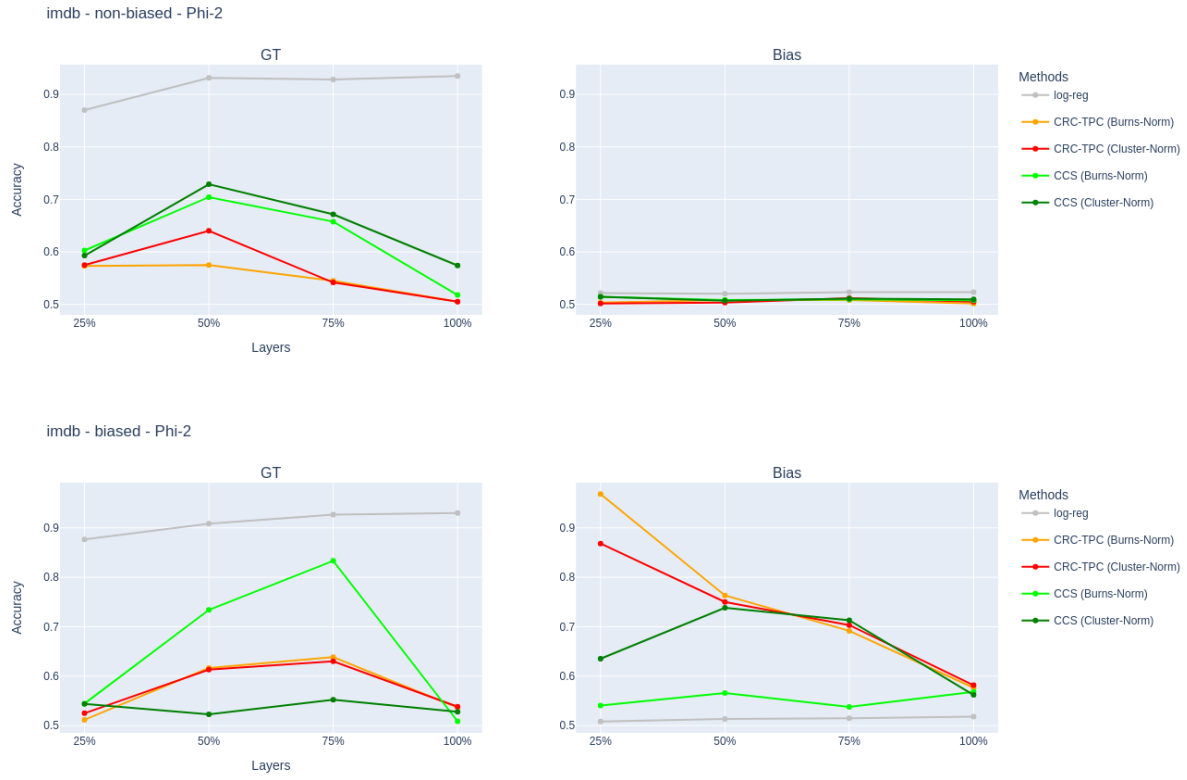

Figure 18: Mean accuracy of Logistic Regression, CRC and CCS probes on Phi-2 on original prompts (up) and biased ones (down) for the explicit opinion experiment.

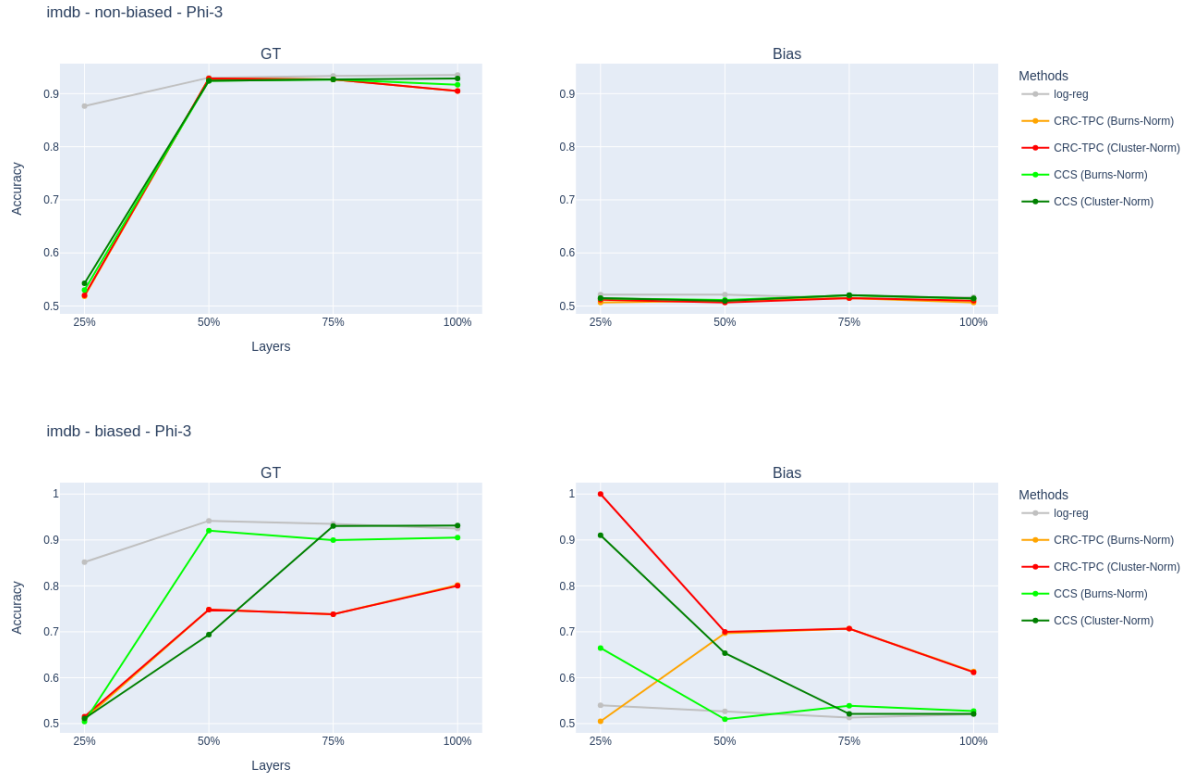

Figure 19: Mean accuracy of Logistic Regression, CRC and CCS probes on Phi-3-Instruct Mini on original prompts (up) and biased ones (down) for the explicit opinion experiment.

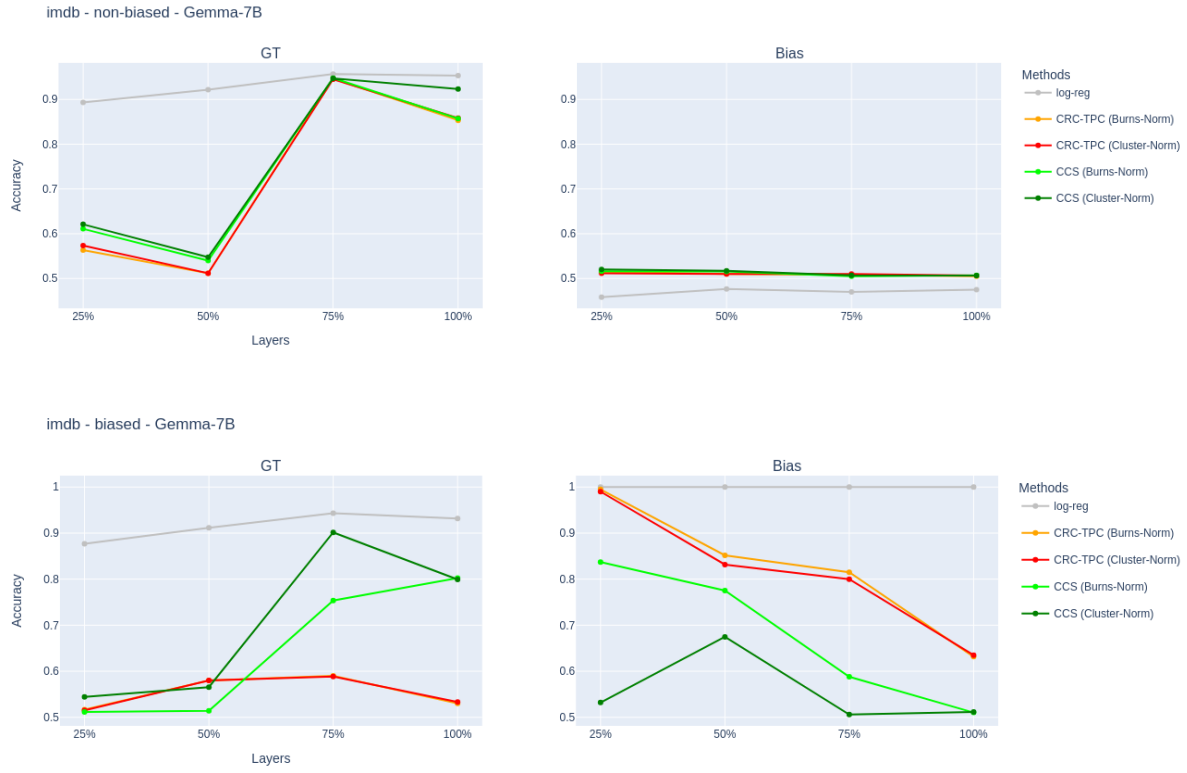

Figure 20: Mean accuracy of Logistic Regression, CRC and CCS probes on Gemma-7B on original prompts (up) and biased ones (down) for the explicit opinion experiment.

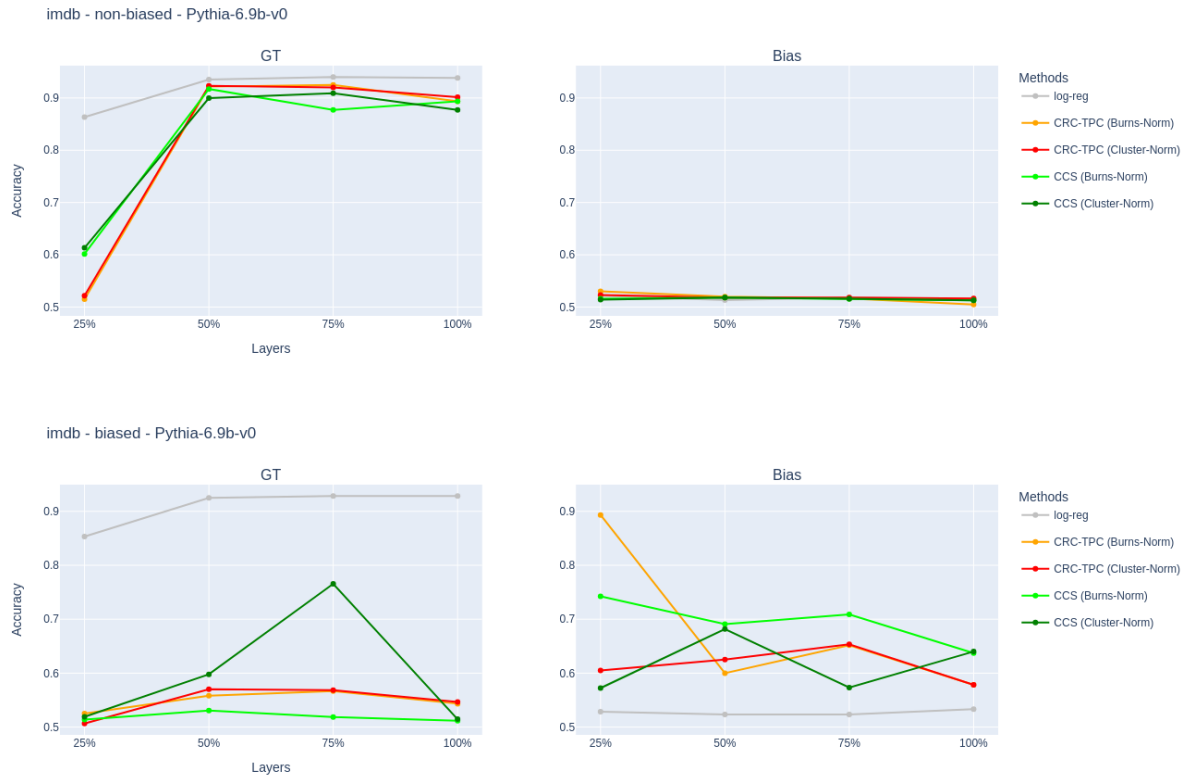

Figure 21: Mean accuracy of Logistic Regression, CRC and CCS probes on Pythia-6.9B-v0 on original prompts (up) and biased ones (down) for the explicit opinion experiment.

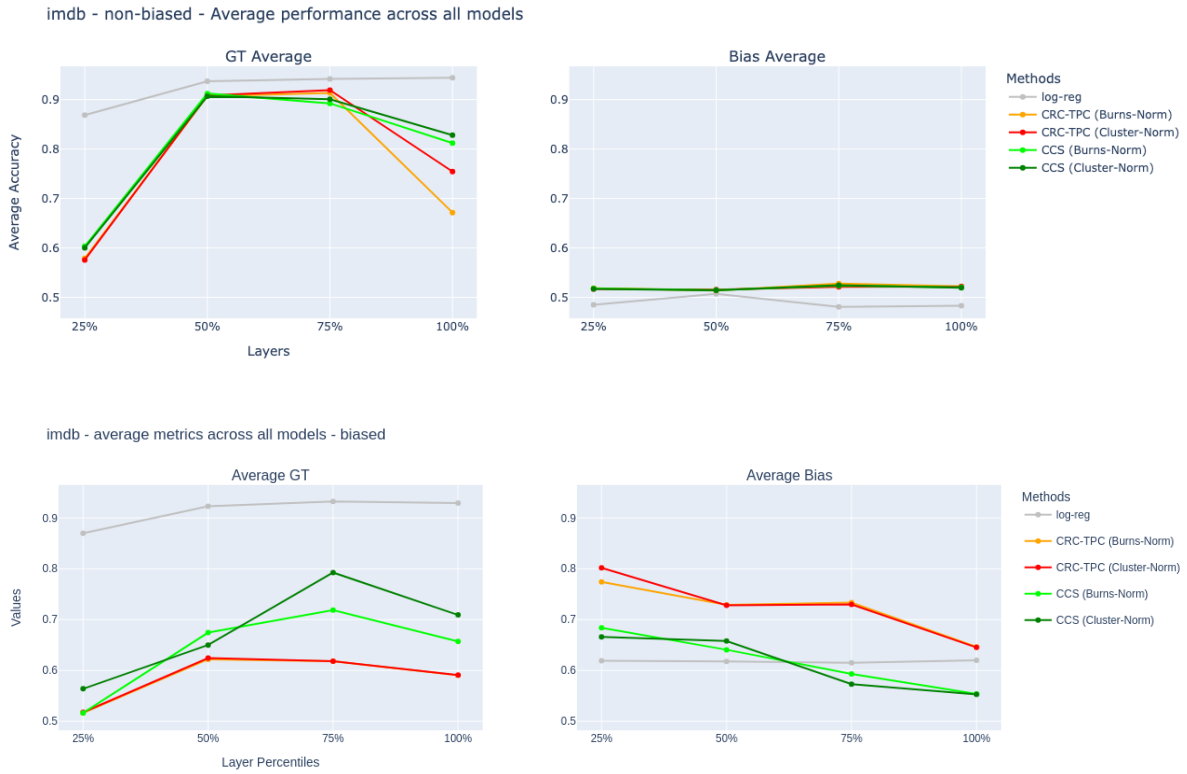

Figure 22: Mean accuracy of Logistic Regression, CRC and CCS probes averaged across all models on original prompts (up) and biased ones (down) for the explicit opinion experiment.

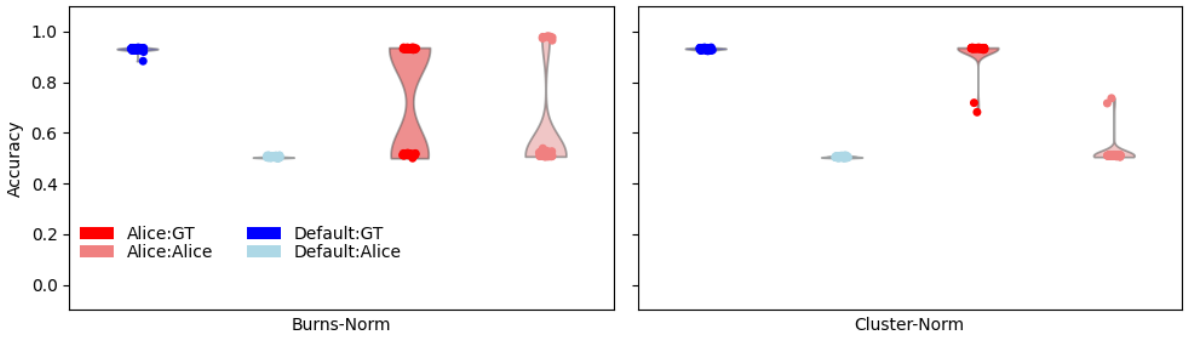

Figure 23: Llama-3-8B - Explicit Opinion Experiment

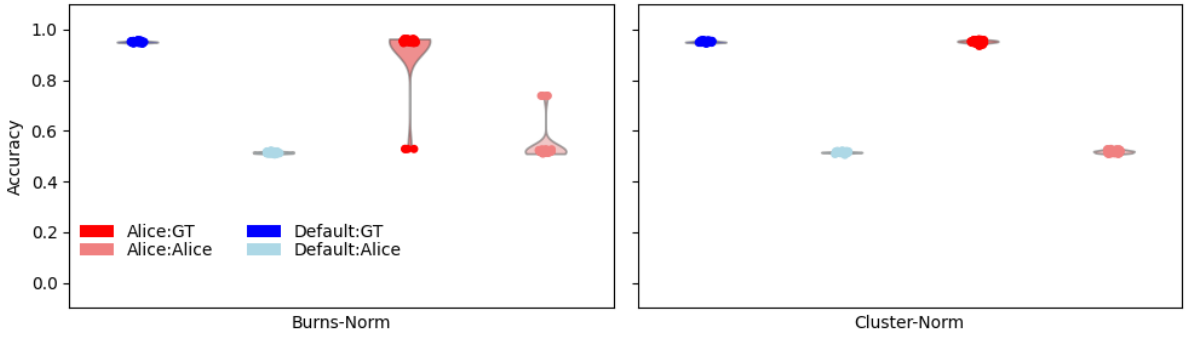

Figure 24: Phi-3 Mini - Explicit Opinion Experiment

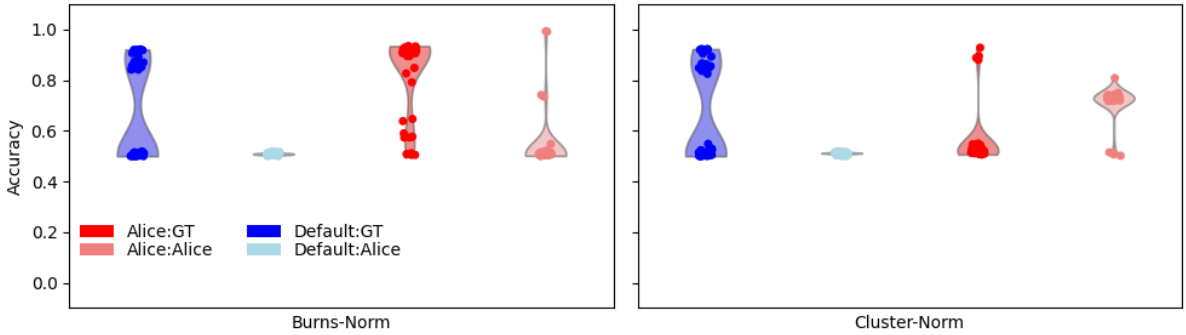

Figure 25: Phi-2 - Compared to Phi-3 and the other models, Phi-2 seems to be an outlier, where probes using Cluster-Norm perform worse than those using Burns-Norms. Possibly, compared to the others, the model is generally less capable.

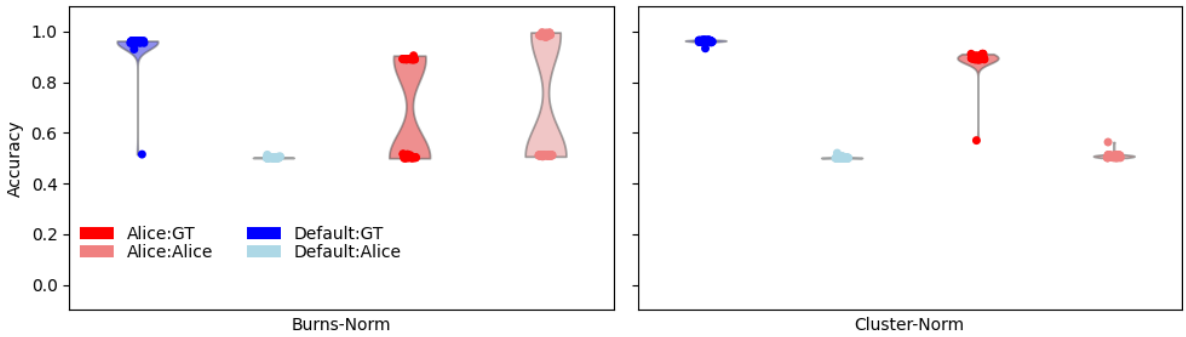

Figure 26: Gemma-7B - Explicit Opinion Experiment

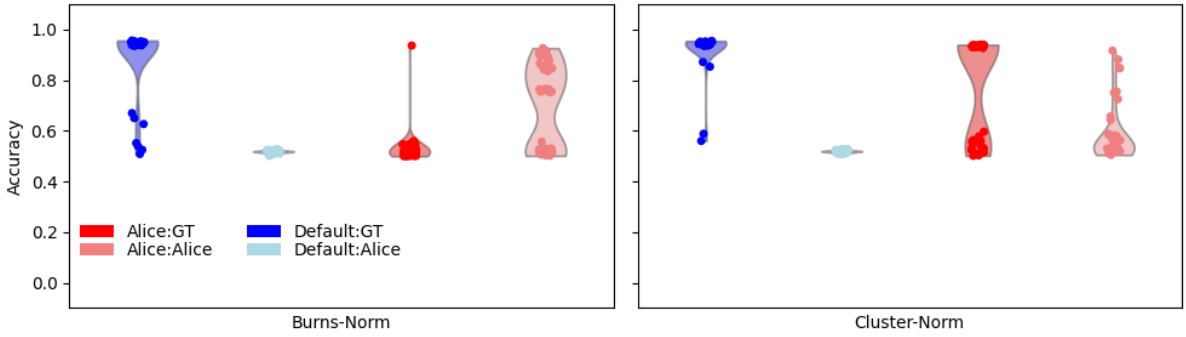

Figure 27: Pythia-6.9B - Explicit Opinion Experiment

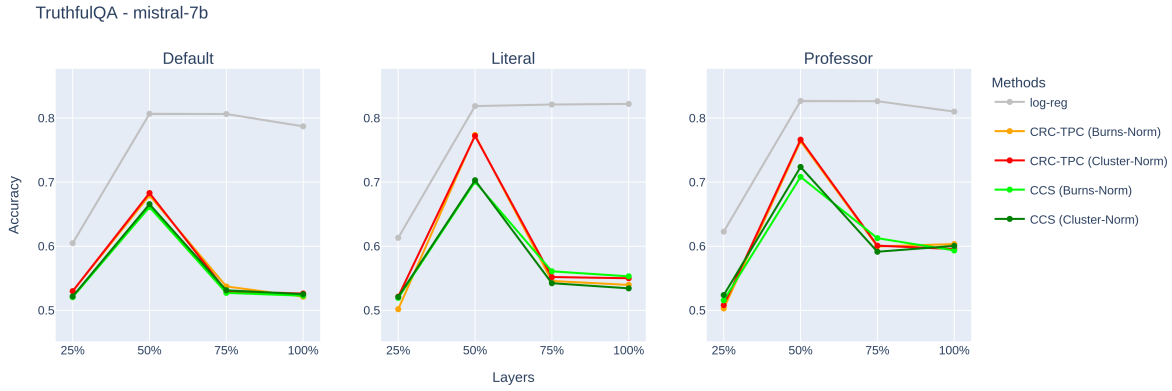

Figure 28: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Mistral-7B, for each prompt template described in Section 4.3.

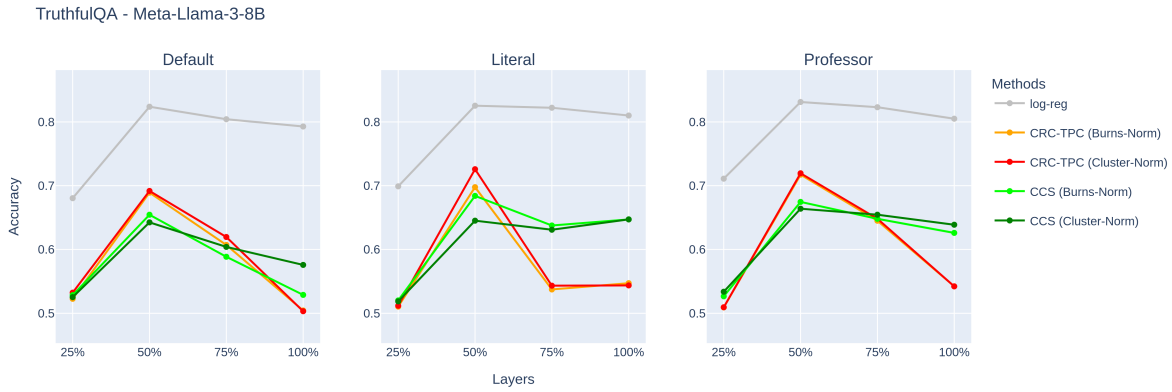

Figure 29: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Llama-3-8B, for each prompt template described in Section 4.3.

Figure 30: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Phi-2, for each prompt template described in Section 4.3.

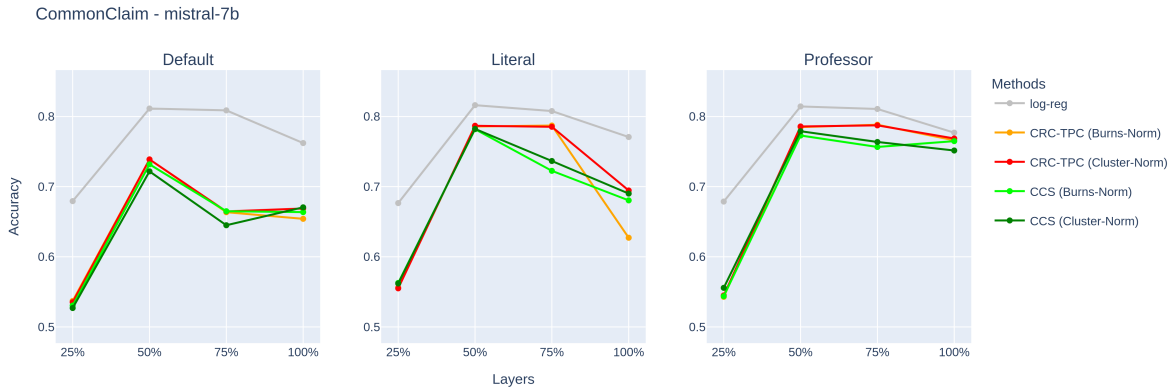

Figure 31: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Mistral-7B, for each prompt template described in Section 4.3.

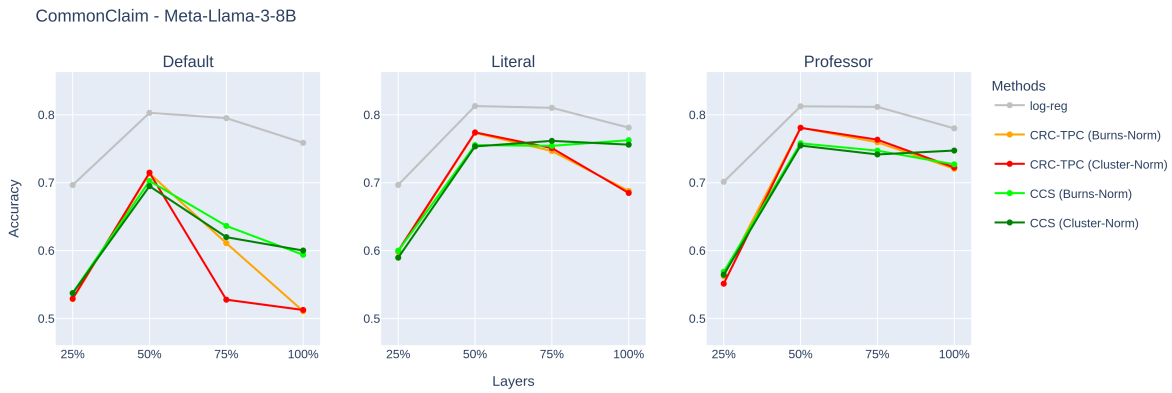

Figure 32: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Llama-3-8B, for each prompt template described in Section 4.3.

Figure 33: Mean accuracy of Logistic Regression, CRC, and CCS probes over 50 probes, using Phi-2, for each prompt template described in Section 4.3.

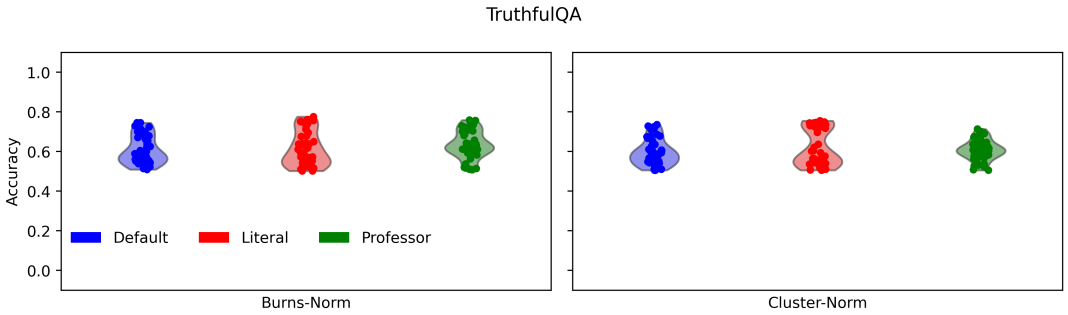

Figure 34: Variation in probe accuracy for the prompt template sensitivity experiment on TruthfulQA for Mistral-7B, at the 75th percentile layer. Contrary to the CommonClaim results (figure 5), variance is too high to be able to conclude anything.

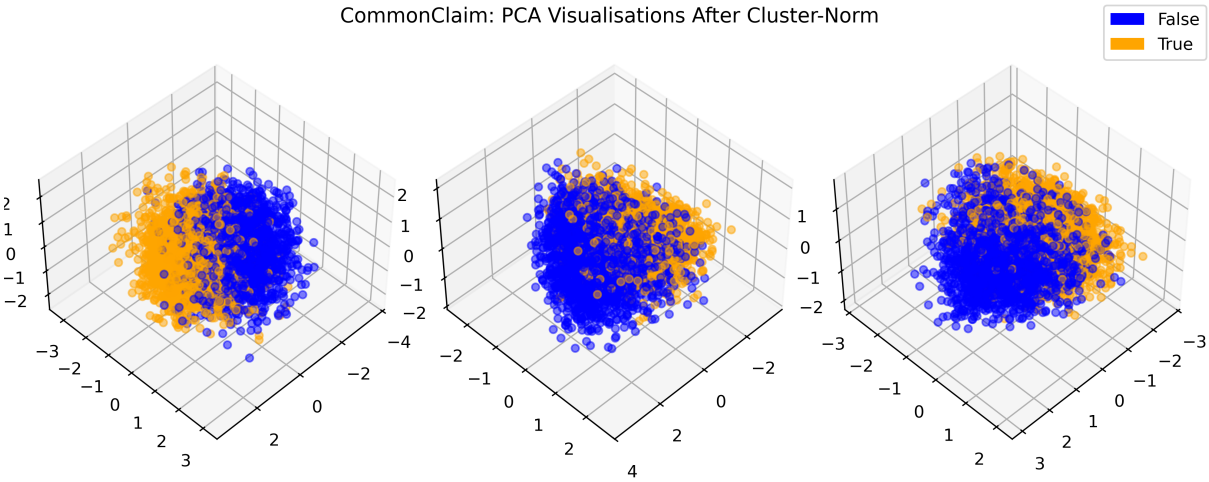

Figure 35: Visualization of the top three PC of $\widetilde{\mathcal{M}}(x_i^+) - \widetilde{\mathcal{M}}(x_i^-)$ - with per cluster normalization - respectively from left to right : for the default, literal and professor prompts. True and False correspond to the ground truth label of these question-answering prompts. We see no notable difference between the three settings, and there is no difference at all to be seen between Burns-Norm and Cluster-Norm. If anything, we can see that in the literal and professor settings, the separation between True and False is slightly more aligned with the first PC.