

Probing Without Labels: Contrast-only Reproductions of Interpretability Results

Stefan F. Schouten
Vrije Universiteit Amsterdam
s.f.schouten@vu.nl

Peter Bloem
Vrije Universiteit Amsterdam
p.bloem@vu.nl

Abstract

Contrastive interpretability methods use pairs (or tuples) of inputs whose elements differ in minimal but controlled ways. Contrast-Consistent Search (CCS) is one such method that can identify whether (language) models encode binary features, such as sentence truth, in their internal activations. One advantage is that this requires no labels, meaning there is no assumption that internal activations agree with those labels, as is the case for supervised methods. In this work, we (1) reproduce results previously obtained with CCS, (2) better understand CCS’s objective, (3) propose Tuple-Contrastive PCA (tcPCA), a method with closed-form solutions, as a spiritual successor, and (4) show how tcPCA has wider applicability than CCS by: allowing for n-ary rather than just binary variables; and, finding contrastive subspaces instead of directions. See contrast.stefanfs.me for code and interactive figures.

1 Introduction

When Large Language Models (LLMs) perform well on benchmarks for a given domain or task, the results are sometimes questioned; in part because of a limited understanding of their working. How is it that LLMs do what they do? Without a clear picture of how an LLM approaches its tasks, we cannot verify if that approach is sensible, or how well it will do outside of benchmarks. One important part of this is identifying what variables are used by a model, where they are encoded, and if they correspond to interpretable *features*.

This paper provides an in-depth look at Contrast-Consistent Search (CCS) (Burns et al., 2023). This unsupervised probing method was introduced to determine if language models represent sentences as true or false. Being unsupervised, it has one advantage: it does not assume that the model’s truth-values agree with human-authored labels. We

perform ablations of the method’s loss terms and find that one of the terms is necessary for a different reason than what originally motivated its inclusion. Based on this insight, we create a method we call Tuple-Contrastive PCA (tcPCA) which performs the same, but yields an entire basis of the latent space, along with interpretable eigenvalues that provide additional insights. We demonstrate that datasets where CCS does not reliably find accurate probes are datasets that fail to isolate a single contrastive feature. We further demonstrate that the method applies to settings where CCS could not (easily) have been applied.

2 Understanding CCS

CCS is a probing method; it uses small classifiers trained on the activations of a larger model to tell if certain information is present. Unlike typical probes, a CCS probe is not given labels to train on. Instead, the probe exploits the fact that its inputs consist of contrastive pairs, which we know to differ along some (set of) feature(s).

Burns et al. (2023) focus on sentences and their negations to train the CCS probes. For example, they used inputs that consisted of question-answer pairs like “Is grass green? Yes/No”, or of declarative sentences such as “Grass is green.” and “Grass is not green.”. Basically, the pair of inputs (X^+ , X^-) consists of language that—if uttered—would amount to an assertion (X^+) or denial (X^-) of a proposition X . The method relies only on the expectation that any latent probability distribution captured by the model’s internals must sum up to one. Of course, we would expect this consistency property for all (binary) variables, not just truth. So in general, we have some binary *feature of interest*, and X^+ and X^- are inputs that primarily (and ideally, exclusively) differ in that feature’s value. Typically, such inputs are minimal pairs where a word is changed, replaced or inserted in order to

Table 1: Accuracy — Mean and standard deviation on 9 datasets, using either *answer* or *period* tokens.

Dataset	<i>answer</i> (%)	<i>period</i> (%)
comparisons	100 ± 00	93 ± 01
sp_en_trans	100 ± 00	95 ± 12
cities	100 ± 00	100 ± 00
amazon	96 ± 00	93 ± 00
imdb	94 ± 00	92 ± 01
ent_bank	84 ± 09	81 ± 19
snli	64 ± 08	82 ± 07
copa	58 ± 06	48 ± 02
rte	54 ± 04	52 ± 03

also change some sentence-level property.

CCS probes operate on mean-centered latent-space activations of X^+ and X^- , which we denote $\mathbf{x}^+$ and $\mathbf{x}^-$, respectively. In this work, we will use CCS with linear probes of the following form: $p(\mathbf{x}) = \sigma(\boldsymbol{\theta}^\top \mathbf{x})$, where $\boldsymbol{\theta}$ are the probe parameters. When using such probes we are assuming that there exists a direction in latent space that the language model uses to represent the feature of interest. By projecting activations on the *feature direction* we can construct a probe that parameterizes a probability distribution for the binary feature of interest.

The objective of CCS consists of a minimization of two terms, the consistency and confidence loss:

$$\begin{aligned}
L_{cons}^\theta(\mathbf{x}^+, \mathbf{x}^-) &= [\sigma(\boldsymbol{\theta}^\top \mathbf{x}^+) - (1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x}^-))]^2 \\
&= [\sigma(\boldsymbol{\theta}^\top \mathbf{x}^+) + \sigma(\boldsymbol{\theta}^\top \mathbf{x}^-) - 1]^2, \\
L_{conf}^\theta(\mathbf{x}^+, \mathbf{x}^-) &= \min\{\sigma(\boldsymbol{\theta}^\top \mathbf{x}^+), \sigma(\boldsymbol{\theta}^\top \mathbf{x}^-), \\
&\quad 1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x}^+), 1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x}^-)\}^2.
\end{aligned}$$

The consistency loss is minimized when the probabilities assigned to sentences and their negations add up to one. The confidence loss is said to prevent the degenerate solution where $p(\mathbf{x}^+) = p(\mathbf{x}^-) = 0.5$. We use the symmetric (unbiased) confidence loss introduced by Farquhar et al. (2023).

We use gradient descent on activations for tokens from X^+ and X^- . For example, with “*Between green and blue, grass is [green/blue]*”, the bracketed tokens could be used for $\mathbf{x}^+$ and $\mathbf{x}^-$.

We begin our analysis of CCS by reporting its performance in different circumstances. We use datasets from three sources (Burns et al., 2023; Marks and Tegmark, 2024; Schouten et al., 2025) (see Appendix D). We include probes trained on both the last and next-to-last tokens, corresponding to a *period* token and an *answer* token. We trained probes for a total of 9 datasets, using 30 different seeds for the probe’s random initialization.

See Appendix C for the full reproduction protocol, and see https://github.com/sfschouten/promise_of_contrast for the code used for all experiments.

In Table 1, we report probe accuracy for layer 16 in Llama-2-7b (Touvron et al., 2023), which is the model we will use throughout the paper. Layer 16 was chosen based on a sweep (see Appendix K). We also ran the experiments for Llama-3.1-8b, the results of which can be found in Appendix M. We find that CCS does not reliably reach well-performing minima for all datasets. Specifically, there are multiple cases where the average performance is above random, but the exact performance varies between seeds (also see Appendix J). We wonder if this could be caused by the two-termed objective, thus our next step is an investigation of what makes the two terms necessary.

2.1 Loss-term ablations

The degenerate solution arises under the following conditions: $\sigma(\boldsymbol{\theta}^\top \bar{\mathbf{X}}^+) = \sigma(\boldsymbol{\theta}^\top \bar{\mathbf{X}}^-) = 0.5 \implies \boldsymbol{\theta}^\top \bar{\mathbf{X}}^+ = \boldsymbol{\theta}^\top \bar{\mathbf{X}}^- = 0$, where $\bar{\mathbf{X}}^+ = [\mathbf{x}_1^+, \mathbf{x}_2^+, \dots, \mathbf{x}_N^+]^\top \in \mathbb{R}^{N \times D}$, is the data matrix for positive samples, and $\bar{\mathbf{X}}^-$ for negative samples. Thus, there are two ways for the degenerate case to arise: (1) the vector learned by the probe has zero length, $|\boldsymbol{\theta}| = 0$; or, (2) the direction points into the null space of the data matrix.¹ To address both paths to the degenerate case, we can alter the CCS training process in two ways.

Alteration 1. By restricting the search space to unit vectors $\hat{\boldsymbol{\theta}}$ we avoid learning the zero vector. Note that the magnitude of the probe parameter vector is unimportant to its accuracy.

Alteration 2. To remove the null space from the data matrix, we use singular value decomposition: $\bar{\mathbf{X}}_{train}^{\pm} = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\bar{\mathbf{X}}_{train}^{\pm} \in \mathbb{R}^{2N \times D}$ is a matrix containing the hidden states of the training data’s positive ($\bar{\mathbf{X}}_{train}^+$), and negative ($\bar{\mathbf{X}}_{train}^-$) samples. We then apply the probes to the reduced representations: $p(\mathbf{x}) = \sigma(\hat{\boldsymbol{\theta}}^\top \mathbf{V}_{:,r}^\top \mathbf{x})$ (with r being the rank of $\bar{\mathbf{X}}_{train}^{\pm}$). This strategy assumes the null spaces of $\bar{\mathbf{X}}_{train}^{\pm}$, $\bar{\mathbf{X}}^+$ and $\bar{\mathbf{X}}^-$ are the same.

Overall, we compare the following kinds of probes: (1) ordinary CCS; (2) only the confi-

¹When training probes, it is not uncommon to work with relatively small datasets. Thus, the number of samples can easily be smaller than the dimensionality of the model’s latent space, resulting in a rank-deficient data matrix.

dence term ($\mathcal{L}_{\text{conf}}$); (3) only the consistency term ($\mathcal{L}_{\text{cons}}$), including versions altered in one ($\mathcal{L}_{\text{cons}}+\text{a1}$, $\mathcal{L}_{\text{cons}}+\text{a2}$) or both ways ($\mathcal{L}_{\text{cons}}+\text{a1}+\text{a2}$); and finally, (4) CCS with only the alterations, no ablations ($\text{CCS}+\text{a1}+\text{a2}$). For this experiment, we use the datasets for which CCS performed well in Table 1.

Results. In Table 2, we give the results of the ablation. We observe that: (1) the ablation of the confidence loss-term reduces accuracy more than the ablation of the consistency loss-term, suggesting the former is more important than the latter; and (2) the proposed alterations do not compensate for the ablation of the confidence loss-term. These results show that the role of the confidence term is not limited to preventing the degenerate solution.

2.2 The effect of the confidence loss

The confidence term encourages probabilities closer to the extremes, but what needs to be true to make that happen? It is minimized when either the positive or negative sample of each pair is assigned a probability of zero or one. But, this would require that $\forall \mathbf{x} : \theta^\top \mathbf{x} = \pm\infty$. Seemingly, minimizing the confidence-loss is just maximizing $\|\theta^\top \bar{\mathbf{X}}\|$ or $\|\theta^\top \mathbf{X}\|$.

When considering unit-length vectors $\hat{\theta}$, the only way to maximize $\hat{\theta}^\top \mathbf{x}$ is to reduce the angle between them. This would mean that the confidence loss is biasing direction $\hat{\theta}$ to have higher variance, i.e. the first (few) principal component(s).

To test this hypothesis, we will compare the directions found by CCS to the principal components. By ablating the two loss terms again, we can see if the confidence loss causes the direction to be more similar to the first few principal components. Specifically, for $\text{CCS}+\text{a1}$, $\mathcal{L}_{\text{conf}}+\text{a1}$, and $\mathcal{L}_{\text{cons}}+\text{a1}$, we will measure: $\lambda^K(\theta) = \frac{1}{\|\theta\|} \|\mathbf{V}_{:K}^\top \theta\|$. This measures how much of θ extends into the subspace spanned by the first K principal components. We apply Alteration 1 to all three objectives. λ^K is a property of the direction alone, but without the unit-length restriction the confidence term can also be lowered by growing $\|\theta\|$ rather than by rotating θ ; restricting to $\hat{\theta}$ leaves rotation toward high-variance directions as the only way to reduce it, which is exactly the effect we want to measure.

Results. In Figure 1, we can see that the confidence-loss indeed causes CCS to find directions closer to the first few principal components of $\bar{\mathbf{X}}$. With only the consistency loss, the

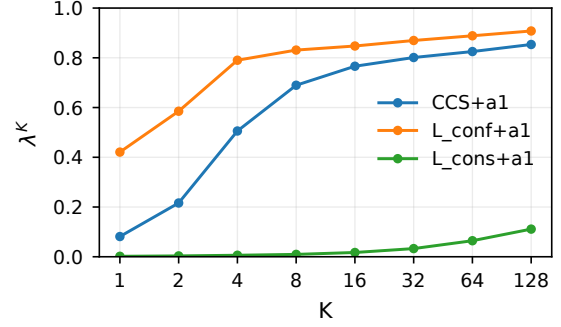


Figure 1: Extent to which learned vectors point into the subspace spanned by the first K principal components. Shown for CCS, only the consistency loss, and only the confidence loss, all with Alteration 1 ($\text{CCS}+\text{a1}$, $\mathcal{L}_{\text{cons}}+\text{a1}$, $\mathcal{L}_{\text{conf}}+\text{a1}$); on the IMDB dataset, using activations for answer tokens, averaged over 30 random seeds.

learned vectors have almost no magnitude in the subspace spanned by the first K principal components. When the confidence loss is added, its magnitude in the high-variance subspace grows. And finally, when we train with only the confidence loss, we find vectors that on average have a cosine similarity of over 0.5 with the first principal component. Results for other datasets and both answer and period token can be found in Appendix E.

Discussion. But why do salient directions make for more accurate CCS probes? We see two main reasons. First, by virtue of how contrastive data are created, the contrastive feature is often (one of) the first principal component(s). This is especially true when using the answer token. This explains why CCS works well in practice. However, the second reason is that we do not care about absolute but *relative* contrast consistency. We have: $\sigma(\hat{\theta}^\top \mathbf{x}^+) + \sigma(\hat{\theta}^\top \mathbf{x}^-) = 1 \implies \hat{\theta}^\top \mathbf{x}^+ = -\hat{\theta}^\top \mathbf{x}^-$, that is, for probabilities to sum to one requires that points be equidistant from the hyperplane orthogonal to $\hat{\theta}$. However, note that finding a direction $\hat{\theta}$ where $\hat{\theta}^\top \mathbf{x}^+ \approx -\hat{\theta}^\top \mathbf{x}^- \implies \hat{\theta}^\top \mathbf{x}^+ + \hat{\theta}^\top \mathbf{x}^- \approx 0$ is not interesting if both sides are already small. In geometric terms, if both points lie very close to the hyperplane, then being near-equidistant in absolute terms is not interesting. What we really care about is $\hat{\theta}^\top \mathbf{x}^+ + \hat{\theta}^\top \mathbf{x}^-$ being small *relative* to the magnitude of both terms. We argue that the confidence loss is an imperfect way of promoting that and instead target relative contrast consistency more directly.

Table 2: Accuracies for probes trained with ablated and/or altered objectives.

$\mu \pm \sigma$ (%)		CCS -	$\mathcal{L}_{\text{conf}}$ -	$\mathcal{L}_{\text{cons}}$ -	$\mathcal{L}_{\text{cons}}$ +a1	$\mathcal{L}_{\text{cons}}$ +a2	$\mathcal{L}_{\text{cons}}$ +a1+a2	CCS +a1+a2
Marks and Tegmark (2024)	comparisons	100±00	99±01	65±12	65±13	67±11	60±07	100±00
	sp_en_trans	100±00	96±08	70±13	70±13	66±10	68±10	100±00
	cities	100±00	97±09	69±11	73±13	69±14	63±10	98±03
Burns et al. (2023)	amazon	96±00	95±00	67±12	70±12	74±13	66±10	96±00
	imdb	94±00	88±06	61±08	61±07	63±10	60±07	94±00

3 Tuple-Contrastive PCA (tcPCA)

Taking CCS as looking for a direction $\hat{\theta}$ along which $|\langle \bar{\mathbf{X}} + \bar{\mathbf{X}} | \hat{\theta} \rangle|$ is small, while $|\langle \bar{\mathbf{X}} | \hat{\theta} \rangle|$ is large, a natural object to look at is $\text{Cov}(\bar{\mathbf{X}} + \bar{\mathbf{X}}) - \alpha \text{Cov}(\bar{\mathbf{X}})$.

By looking at the *difference in variance* between the pairwise sums $\text{Cov}(\bar{\mathbf{X}} + \bar{\mathbf{X}})$ and the total $\text{Cov}(\bar{\mathbf{X}})$, we capture the need for relative contrast consistency. $\text{Cov}(\bar{\mathbf{X}})$ captures both intra- and inter-pair variance, but the pairwise sum destroys the intra-pair variance. Thus, the difference between them captures exactly what we want.

Note that:

$$\begin{aligned} \text{Cov}(\bar{\mathbf{X}} + \bar{\mathbf{X}}) &= \text{Cov}(\bar{\mathbf{X}}) + \text{Cov}(\bar{\mathbf{X}}) \\ &\quad + \text{Cov}(\bar{\mathbf{X}}, \bar{\mathbf{X}}) + \text{Cov}(\bar{\mathbf{X}}, \bar{\mathbf{X}}). \end{aligned}$$

And, with $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}$ mean-centered independently, we have $\text{Cov}(\bar{\mathbf{X}}) + \text{Cov}(\bar{\mathbf{X}}) = 2 \text{Cov}(\bar{\mathbf{X}})$, thus:

$$\begin{aligned} \text{Cov}(\bar{\mathbf{X}} + \bar{\mathbf{X}}) - 2 \text{Cov}(\bar{\mathbf{X}}) \\ = \text{Cov}(\bar{\mathbf{X}}, \bar{\mathbf{X}}) + \text{Cov}(\bar{\mathbf{X}}, \bar{\mathbf{X}}) \end{aligned}$$

This means that the *difference in variance* involves only the cross-covariance terms, as long as we pick $\alpha = 2$. This analysis is essentially identical to that of Belrose et al. (2024), who analyze CRC-TPC, rather than CCS.

The cross-covariance terms tell us how correlated the activations of the elements of pairs are, i.e. within each pair, along a given direction, do both elements have the same activations, opposite activations, or somewhere in between. Directions with (near) perfect correlation are uninteresting, as those are directions which evidently do not encode whatever is different within the pairs.

By taking the eigendecomposition of $\text{Cov}(\bar{\mathbf{X}} + \bar{\mathbf{X}}) - 2 \text{Cov}(\bar{\mathbf{X}})$, we get a basis over our latent space whose axes are ordered by their relative contrast consistency. We call this approach Tuple-Contrastive PCA, because of its connections to

Contrastive PCA (Abid et al., 2017) specifically, and PCA in general (see Appendix B). We refer to the basis vectors as tuple-contrastive Principal Components (tcPCs).

We can generalize this method to tuples of inputs, with activation matrices $\langle \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \rangle$, by using $\text{Cov}(\sum_i \mathbf{X}^{(i)}) - N \text{Cov}(\mathbf{X}^*)$, where $\mathbf{X}^*$ concatenates each matrix like $\bar{\mathbf{X}}$ before.

This allows us to use the method also for non-binary features, as we will demonstrate (also see Appendix A).

Generalized tcPCA (g-tcPCA) We also study a version where we eigendecompose $\text{Cov}(\sum_i \mathbf{X}^{(i)})$ after applying a transformation that whitens the data matrix $\mathbf{X}^*$. This is equivalent to solving the generalized eigenproblem:

$$\text{Cov}\left(\sum_i \mathbf{X}^{(i)}\right) \mathbf{v} = \lambda \text{Cov}(\mathbf{X}^*) \mathbf{v}$$

which gives eigenvalues:

$$\lambda = \frac{\mathbf{v}^T \text{Cov}(\sum_i \mathbf{X}^{(i)}) \mathbf{v}}{\mathbf{v}^T \text{Cov}(\mathbf{X}^*) \mathbf{v}}.$$

That is, it divides where tcPCA subtracts, and it gives us a ratio telling us by what factor the summation caused the variance to shrink. This variant is related to Generalized Contrastive PCA (Oliveira et al., 2025; Woller et al., 2025).

3.1 Binary Accuracy

Accuracies for classifying with the top tcPC can be seen in Table 3. It is clear that when CCS converges to the same performance reliably, tcPCA matches that performance almost exactly. And, for those datasets where CCS is sensitive to the random initialization, tcPCA has accuracies somewhere between the minimum and the maximum of CCS. Another method proposed by Burns et al. (2023) is CRC-TPC, which simply takes the top principal component of $\bar{\mathbf{X}} - \bar{\mathbf{X}}$. It can be seen to perform considerably worse on the period token for three

Table 3: Accuracy on *period* token of tcPCA and g-tcPCA compared to CCS. Bold indicates results that match or exceed CCS median and CRC-TPC.

Dataset	CCS			CRC -TPC	tcPCA	g-tcPCA
	min	med	max			
comparisons	93	93	94	52	94	96
sp_en_trans	56	99	99	99	99	99
cities	100	100	100	51	99	100
amazon	92	93	93	51	93	93
imdb	90	92	93	91	92	93
ent_bank	50	93	96	89	90	89
snli	68	84	92	75	81	84
copa	44	48	55	50	51	52
rte	47	52	59	52	52	52

datasets, likely because it finds a direction with high overall variance instead of tuple-contrastive variance (see Appendix F).

3.2 Interpreting Eigenvalues

One potential problem with contrast-based probing is that even if we construct the probing data ourselves, it is hard to be 100% sure that we have isolated a single feature. Not only can features be hard to differentiate from each other, but an LLM may encode matters in a way that does not match our understanding of the problem. Looking at the distribution of tcPC eigenvalues can be of help. If the contrasting data and the model’s encoding thereof meets our expectations, then the first eigenvalue(s) should stand out from the rest. For a single binary feature, we expect to see one and only one direction as clearly contrast consistent.

In Figure 2, we can see the top-10 eigenvalues for different datasets (see Appendix H for all datasets). Looking at, for example, the *amazon* dataset we see precisely what we wanted; the first eigenvalue is clearly largest. Going to *copa*, we see a somewhat flatter distribution of eigenvalues. For this dataset, there is a second stand-out tcPC which we will look at in more detail. And for *snli*, we can see an even more diffuse distribution. Moreover, we consistently see more diffuse eigenvalues precisely in those cases where CCS’s performance is less reliable. Specifically, over the 18 dataset $\times$ token combinations, the Spearman ρ calculated between (1) the eigen-gap λ_0/λ_1 , and (2) the variance in CCS accuracy induced by random initialization is -0.62 for Llama-2-7b and -0.82 for Llama-3.1-8b. Figure 3 shows this relation for Llama-2-7b (see Appendix I for Llama-3.1-8b).

Case study: COPA. Choice of Plausible Alternatives (Roemmele et al., 2011) is a commonsense causal reasoning dataset. It consists of prompts such as: *Consider the following example: “The bar closed.” Choice 1: It was crowded. Choice 2: It was 3 AM. Q: Which is more likely to be the cause, choice 1 or choice 2? choice [1/2].*

None of the approaches perform well on this dataset, typically doing no better than random guessing. For the answer token, CCS does occasionally find directions that perform better at 69%, suggesting that the model is not at fault. Given that the two highest eigenvalues seem to stand out among the rest, it makes sense to ask: (1) what does the first tcPC represent, if not ‘sentence truth’? and (2) does the second tcPC encode ‘sentence truth’?

To answer the first question, we looked at a subset of COPA together with the relevant activations, i.e. the projections of the answer token hidden states onto the first tcPC. Looking at the most contrastive examples (where activations had the highest absolute values) quickly revealed the answer. It appears that in COPA, ‘sentence truth’ is not the only thing that changes between positive and negative samples, the answers often also differ in sentiment. In Table 4, we can see some examples of how high activations (left) correspond to the occurrence of comparatively ‘bad’ situations or events in answers (negative sentiment) and low activations (right) correspond to comparatively ‘good’ situations or events (positive sentiment). See Appendix G for full samples.

As for the second question, the answer is likely yes. The second tcPC predicts ‘sentence truth’ on COPA with 70% accuracy, against CCS’s 69% maximum out of 30 runs.

3.3 Negation and Truth

The ability to find multiple directions can also be used deliberately. In recent work, Bürger et al. (2024) show that negation and truth occupy a shared subspace. To showcase the utility of our approach in a multivariate setting, we will replicate their results. We use the *cities* dataset, varying both the negation and the country that a city is said to lie in. For a given city, we have a tuple: $\langle \mathbf{x}^{p,c}, \mathbf{x}^{p,i}, \mathbf{x}^{n,c}, \mathbf{x}^{n,i} \rangle$, where p and n indicate positive and negative polarity (negation), and c and i indicate the correct and incorrect country. Of these, only $\mathbf{x}^{p,c}$ (affirmation of correct country) and $\mathbf{x}^{n,i}$ (denial of incorrect country) are true statements.

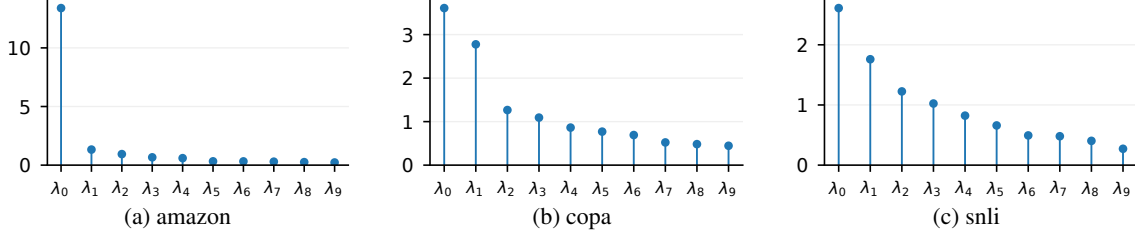


Figure 2: Top eigenvalues for three datasets with different eigendistributions. Based on *answer* tokens.

Table 4: COPA samples, showing the prompt and cause/effect options along with their activation strength a on the contrastive feature encoded by the top tcPC. The choices labeled as commonsense are underlined.

	prompt	a negative sentiment	a positive sentiment
effect	‘The scientist conducted an experiment.’	3.7 ‘She fabricated her data.’	-2.7 ‘ <u>She validated her theory.</u> ’
	‘The man turned on the faucet.’	2.7 ‘The toilet filled with water.’	-3.5 ‘ <u>Water flowed from the spout.</u> ’
cause	‘The computer crashed.’	3.3 ‘ <u>I downloaded a virus.</u> ’	-3.1 ‘I backed up my files.’
	‘I learned how to play the board game.’	3.1 ‘My friend got the rules wrong.’	-2.6 ‘ <u>My friend explained the rules to me.</u> ’

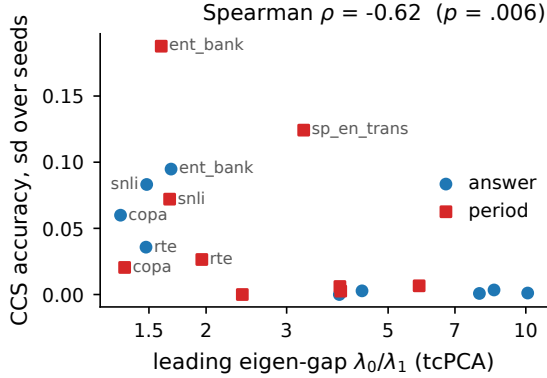


Figure 3: SD for CCS accuracy over 30 seeds against leading tcPCA eigen-gap λ_0/λ_1 , one point per dataset and token (Llama-2-7B).

Results. In Figure 4, we can see activations for the portion of the dataset we held out for evaluation, plotted by their coordinates along the first, second and third g-tcPCA components (see Figure 13 for tcPCA). We see clear separation by: the statement’s *truth*, whether the statement would have been true had it not been negated (*base*), and *negation*. In fact, what we see is a tetrahedron, which appears like three separate rectangular arrangements of the data, depending on which side you view it from. This not only replicates, but extends the results of Bürger et al. (2024), showing the activations are arranged by a third axis along the *negation* feature.

3.4 Non-binary features

We also reproduce two results which probe for non-binary features. First, Savcicens and Eliassi-Rad

(2025) probe for trinary truth values: True, False, Neither. They use a few datasets, among which is the *cities* dataset modified such that it also includes sentences with non-existent city names, thereby creating the Neither inputs. We use this dataset to create tuples $\langle \mathbf{x}^t, \mathbf{x}^f, \mathbf{x}^n \rangle$.

Second, Engels et al. (2025) search for features which are not represented along one dimension, but multiple. For example, they find that LLMs represent ‘days of the week’ (and ‘months of the year’) in a circular representation. We create tuples which consist of inputs differing only by the day of the week that they mention $\langle \mathbf{x}^{d_1}, \dots, \mathbf{x}^{d_7} \rangle$ (see Appendix D).

Results. Both reproductions succeed. For the trinary truth values, we see that the model itself also organizes truth values using a triangle, the same shape Savcicens and Eliassi-Rad (2025) used to visualize their results (Figure 5).

The ‘days of the week’-experiment also reproduces with a circular structure being present that encodes the days in the correct order (Figure 6).

3.5 Discussion

The findings on which the reproductions in Sections 3.3 and 3.4 are based originally all required specialized probing methodology. We reproduce them with a conceptually simple and general-purpose method, requiring only the contrastive data, which was either already present or trivial to create.

Our findings show that tcPCA is a valuable interpretability tool, taking the strengths of PCA but using it to isolate only the variance that exists within

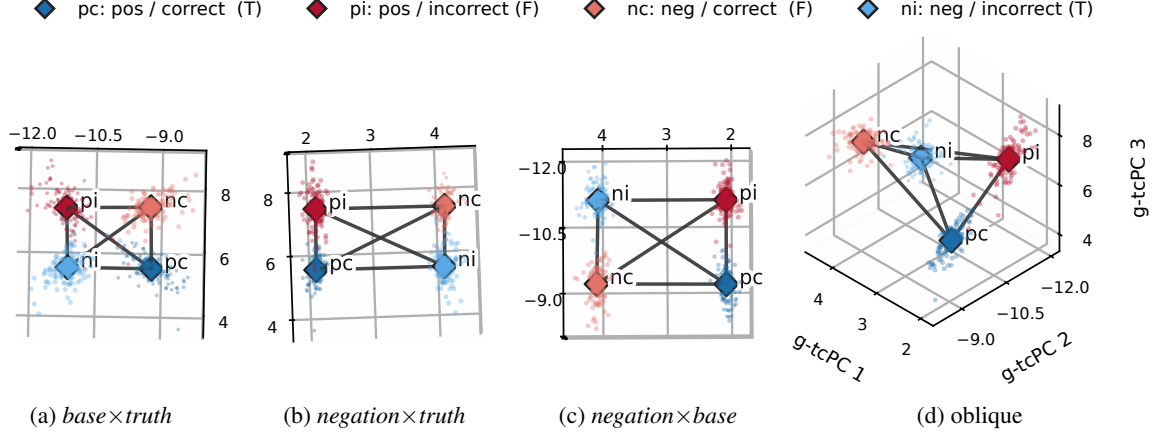


Figure 4: The four statement types $\langle \mathbf{x}^{p,c}, \mathbf{x}^{p,i}, \mathbf{x}^{n,c}, \mathbf{x}^{n,i} \rangle$ on the first three g-tcPCA components (Llama-2-7b). (a)–(c) look down the axis that collapses one contrast. tcPCA views in Appendix L.

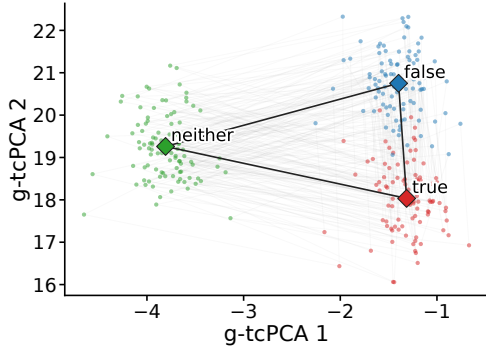


Figure 5: Projections of $\langle \mathbf{x}^t, \mathbf{x}^f, \mathbf{x}^n \rangle$ (red, blue, green, respectively) onto first and second g-tcPCA components. Grey lines show contrast-pairs.

tuples. As a tool, it occupies an underexplored niche: directed but unsupervised. This places it somewhere in between traditional supervised probing methods, which are directed with supervision, and typical unsupervised methods, which are undirected. By deciding on the difference between the elements of each tuple, we can direct tcPCA to specific features, while remaining unsupervised, i.e. it does not optimize for classification accuracy. Instead, it allows us to find and identify both the feature and its correlates, rather than finding some linear combination that performs well on the probe training data.

4 Conclusion

We have reproduced results previously obtained with CCS, and observed that it does not reliably converge to an accurate classifier. By performing an ablation we re-understand the purpose of the loss terms, which led us to propose a spiritual suc-

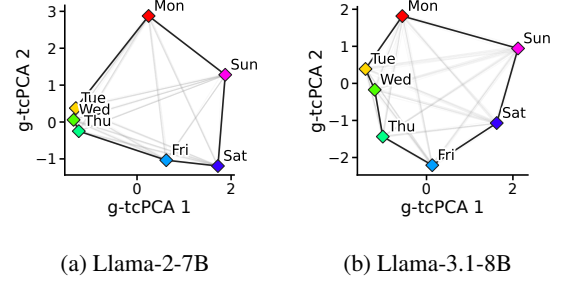


Figure 6: Projections of $\langle \mathbf{x}^{d_1}, \dots, \mathbf{x}^{d_7} \rangle$ onto first and second g-tcPCA components. Grey lines show contrast-pairs. Layer 16 for both models.

cessor we call tcPCA, providing a full basis over the latent space in closed form. It shows that the pairwise differences in some datasets do not isolate a single direction in latent space, explaining why CCS does not always converge to the same solution. Specifically, across datasets, the smaller the leading eigen-gap, the more CCS varies between seeds. A generalized variant, g-tcPCA, divides where tcPCA subtracts, which makes its eigenvalues independent of the scale of each direction. We have also shown that tcPCA, unlike CCS, can be used to find contrastive subspaces of more than one dimension, improving its applicability. Using tuples of more than two inputs, it recovers structures without supervision: truth and negation arranged as a tetrahedron; true, false and neither as a triangle; and the days of the week as a circle. Most findings carry over to a second model, Llama-3.1-8B.

Future work should further investigate tcPCA, its applications, and whether it can live up to its promise as a directed but unsupervised interpretability method.

Limitations

We used two models from the same family, with Llama-2-7b being used for the results presented in the main text, and Llama-3.1-8B being presented in Appendix M. We report results for one layer at a time, but a layer sweep motivating the choice is available in Appendix K.

The COPA dataset turned out to have a clear second feature whose value differs between each pair, but for the datasets with even more diffuse eigenvalue distributions it was not as easy to identify what human-comprehensible feature (if any) was encoded along those directions.

Acknowledgments

This research was supported by Huawei Finland through the DreamsLab project. All content represented the opinions of the authors, which were not necessarily shared or endorsed by their respective employers and/or sponsors.

We would like to sincerely thank the reviewers for their great feedback and numerous suggestions, which included the idea for a 3-dimensional visualization.

References

- Abubakar Abid, Martin J. Zhang, Vivek K. Bagaria, and James Zou. 2017. [Contrastive Principal Component Analysis](#). *arXiv preprint*. ArXiv:1709.06716 [stat].
- Nora Belrose, Alex Mallen, Dhruva Ghosh, Walter Lauer, Kyle O’Brien, Alexander Wan, Ben Wright, Akari Asai, and Yanai Elazar. 2024. [VINC-S: Closed-form Optionally-supervised Knowledge Elicitation with Paraphrase Invariance](#).
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. [Discovering Latent Knowledge in Language Models Without Supervision](#). In *11th International Conference on Learning Representations (ICLR 2023)*, pages 18404–18431. Curran Associates, Inc.
- Lennart B rger, Fred A. Hamprecht, and Boaz Nadler. 2024. [Truth is Universal: Robust Detection of Lies in LLMs](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 138393–138431. Curran Associates, Inc.
- Josh Engels, Eric Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2025. [Not All Language Model Features Are One-Dimensionally Linear](#). *International Conference on Learning Representations*, 2025:84591–84622.

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 [cs].

Samuel Marks and Max Tegmark. 2024. [The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets](#). Philadelphia, Pennsylvania, United States.

Eliezyer Fermino de Oliveira, Pranjal Garg, Jens Hjerling-Leffler, Renata Batista-Brito, and Lucas Sjulson. 2025. [Identifying patterns differing between high-dimensional datasets with generalized contrastive PCA](#). *PLOS Computational Biology*, 21(2):e1012747.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. [Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning](#). Stanford University.

Germans Savcisens and Tina Eliassi-Rad. 2025. [Trilemma of Truth in Large Language Models](#).

Stefan F. Schouten, Peter Bloem, Ilia Markov, and Piek Vossen. 2025. [Truth-value judgment in language models: ‘truth directions’ are context sensitive](#). Montreal, Quebec, Canada.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open Foundation and Fine-Tuned Chat Models](#). *arXiv preprint*. ArXiv:2307.09288 [cs].

Joshua P. Woller, David Menrath, and Alireza Gharabaghi. 2025. [Generalized contrastive PCA is equivalent to generalized eigendecomposition](#). *PLOS Computational Biology*, 21(10):e1013555.

A Generalization to non-binary variables

When generalizing to non-binary variables, each variable can take more than two values. Thus, we consider not pairs $\langle \bar{\mathbf{X}}, \bar{\mathbf{X}} \rangle$, but tuples $\langle \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \rangle$. A row from each of those matrices yields $\langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \rangle$, which correspond to activations collected for inputs from N related inputs, with each input differing from the others in a minimal way. We then have a tuple-wise summation for which the covariance breaks down as

follows:

$$\begin{aligned} \text{Cov}\left(\sum_i \mathbf{X}^{(i)}\right) \\ = \sum_{i \neq j} \text{Cov}(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}) + \sum_i \text{Cov}(\mathbf{X}^{(i)}) \end{aligned}$$

The second term is now equivalent to N times the covariance of a matrix $\mathbf{X}^*$ which contains all the rows of every $\mathbf{X}^{(i)}$. To see this, center each $\mathbf{X}^{(i)}$ separately, so that $\mathbf{X}^*$ has zero mean. Then, with n tuples, $\text{Cov}(\mathbf{X}^*) = \frac{1}{Nn} \sum_i \mathbf{X}^{(i)\top} \mathbf{X}^{(i)} = \frac{1}{N} \sum_i \text{Cov}(\mathbf{X}^{(i)})$, and hence $N \text{Cov}(\mathbf{X}^*) = \sum_i \text{Cov}(\mathbf{X}^{(i)})$.

When we subtract that second term, we are once again left with only the cross-covariance terms. That means that when we take the eigendecomposition we now look for directions along which activations for the inputs of each feature value are as un- or anti-correlated between each other as possible.

B tcPCA and Contrastive PCA

Contrastive PCA (Abid et al., 2017) is a variant of PCA where we find directions that correspond to maximal differences between the variance of some data matrix of *target* samples ($\mathbf{T}$) and the variance of a matrix of *background* samples ($\mathbf{B}$):

$$\arg \max_{\hat{\theta}} \hat{\theta}^\top \text{Cov}(\mathbf{T}) \hat{\theta} - \alpha \hat{\theta}^\top \text{Cov}(\mathbf{B}) \hat{\theta} \quad (1)$$

Tuple-Contrastive PCA is thus an instance of Contrastive PCA where we choose the target and background distributions in a particular way. Given a set with vector-valued tuples, we choose (1) the tuple-wise sums ($\sum_i \mathbf{X}^{(i)}$), and (2) the concatenation of vectors from every position in each tuple ($\mathbf{X}^*$), as the target and background distributions.²

With $\alpha = N$, and with each element of the tuples centered separately, the two variance terms cancel and only the cross-covariance terms remain (Appendix A); this is the tcPCA objective. Generalized tcPCA (§3) relates in the same way to generalized contrastive PCA (Oliveira et al., 2025; Woller et al., 2025), which replaces the difference of the two variances by their ratio and so needs no α .

²Which is which only changes the interpretation of α , and whether we are ultimately interested in the top tcPCs or the bottom tcPCs.

C Reproduction protocol

We use Llama-2-7B³ and Llama-3.1-8B⁴ in their native precision (fp16 and bf16, respectively), and take the residual stream after decoder block 15 (i.e. layer 16) at the second-to-last token (*answer*) and the last token (*period*).

The pairs of each dataset are split 50/50 into training and evaluation pairs, with the same split for every method and seed, and the activations of the two elements of the pairs are mean-centered separately.

CCS is trained with AdamW (learning rate 10^{-3} , weight decay 10^{-2}) for 1000 epochs from each of 30 random initializations, without normalizing the variance of the activations. Because the directions found by CCS have no inherent sign, we orient each one to point from the mean of the false towards the mean of the true training statements. A pair counts as correct when the prediction $\frac{1}{2}(p(\mathbf{x}^+) + 1 - p(\mathbf{x}^-)) > 0.5$ agrees with its label; we use no calibration.

tcPCA is the exact eigendecomposition of $\text{Cov}(\sum_i \mathbf{X}^{(i)}) - N \text{Cov}(\mathbf{X}^*)$; for g-tcPCA we add a ridge of $10^{-8} \|\text{Cov}(\mathbf{X}^*)\|_F$ to $\text{Cov}(\mathbf{X}^*)$ before solving the generalized eigenproblem.

Code and data availability. All code is available at https://github.com/sfschouten/promise_of_contrast. A single command (`dvc repro check_paper`) should regenerate every table and figure. For each dataset, the repository lists the exact examples used for training and for evaluation. Interactive versions of the figures are available at <https://contrast.stefanfs.me>.

D Datasets

D.1 Prompts used for original CCS datasets

D.1.1 amazon

```
{
  answer_choices: "negative ||| positive",
  jinja: "Consider the following example: '"
    '{{content}}' Between
    '{{answer_choices[0]}} and
    '{{answer_choices[1]}}', the sentiment of
    this example is |||
    '{{answer_choices[label]}}'"
}
```

D.1.2 imdb

³meta-llama/Llama-2-7b-hf

⁴meta-llama/Meta-Llama-3.1-8B

```
{
  answer_choices: "negative ||| positive",
  jinja: "The following movie review expresses
    what sentiment? {{text}} |||
    {{answer_choices[label]}}"
}
```

D.1.3 copa

```
{
  answer_choices: "choice 1 ||| choice 2",
  jinja: "Consider the following example:
    '{{premise}}' Choice 1: {{choice1}}
    Choice 2: {{choice2}} Q: Which one is
    more likely to be the {{question}}, choice 1
    or choice 2? ||| {{answer_choices[label]}}"
}
```

D.1.4 rte

```
{
  answer_choices: "incorrect ||| correct",
  jinja: 'Assuming that the following is true:
    "{{text1}}" \n Concluding that: "{{text2}}"
    is ||| {{answer_choices[label]}}'
},
```

D.2 Trinary truth values

We used the dataset from [hf.co/datasets/carlmarmarxx/trilemma-of-truth](https://huggingface.co/datasets/carlmarmarxx/trilemma-of-truth) to construct the triples.

D.3 Days-of-the-week feature

Like [Engels et al. \(2025\)](#), we use the following prompts.

Let's do some day of the week math. NUMBER days from DAY is

Where DAY is replaced with one of the weekdays, and NUMBER is chosen from {"Zero", "One", ..., "Twenty"}.

E Maximum variance effect of confidence-loss

The effect of the confidence loss in all settings can be seen in Figure 7. The pattern of Figure 1 holds. With only the consistency term, the learned direction stays almost entirely outside the leading principal subspace. With the confidence term, alone or in CCS, it lies largely inside it. The confidence term concentrates the direction in the very first components.

F Performance of CRC on top 5 contrastive principal components

The performance of the principal components of $\bar{\mathbf{X}} - \bar{\mathbf{X}}^+$ can be seen in Table 5. When we use the top

principal component this is equivalent to CRC-TPC ([Burns et al., 2023](#)).

On the *answer* token the first component of $\bar{\mathbf{X}} - \bar{\mathbf{X}}^+$ is accurate on most datasets, as CCS is. On the *period* token it drops to chance on comparisons, cities and amazon, where the direction that does separate true from false appears further down the list of principal components.

G Full COPA examples

The set of COPA examples on which our analysis of the tuple-contrastive Principal Components is based can be seen in Table 6. They are held-out datapoints whose two choices the first tcPC separates most (top) and least (bottom).

H Eigenvalue Distributions

Figure 8 shows the eigenvalue distributions on all datasets for the answer token.

I Eigenvalue spectra and CCS reliability

§3.2 observes that datasets with a diffuse tcPCA spectrum are those on which CCS is unreliable. Figure 9 quantifies this. For every dataset and token it plots the standard deviation of CCS accuracy over 30 seeds against the leading eigen-gap λ_0/λ_1 . The two are strongly rank-correlated on both models: Spearman $\rho = -0.62$ ($p = .006$) on Llama-2-7B and $\rho = -0.82$ ($p < .001$) on Llama-3.1-8B, over 18 dataset-token combinations each.

J CCS accuracy per seed

Figure 10 shows the accuracy of each of the 30 CCS seeds behind Table 1.

K Choice of layer

We use layer 16 (the output of decoder block 15) unless stated otherwise. To check that this choice does not favor any method, we fit tcPCA, g-tcPCA, CCS (5 seeds) and a supervised logistic probe at every decoder block from 8 to 24, with the same protocol as the main results. Figure 11 shows the accuracy averaged over the nine datasets. Every method plateaus from block 14 to 18 on Llama-2-7B (14 to 17 on Llama-3.1-8B), and on both models and both tokens block 15 is within 1.3 points of each method’s best layer.

For the figures of non-binary features (days of the week, true/false/neither) we choose the layer per figure, by how cleanly the classes separate in

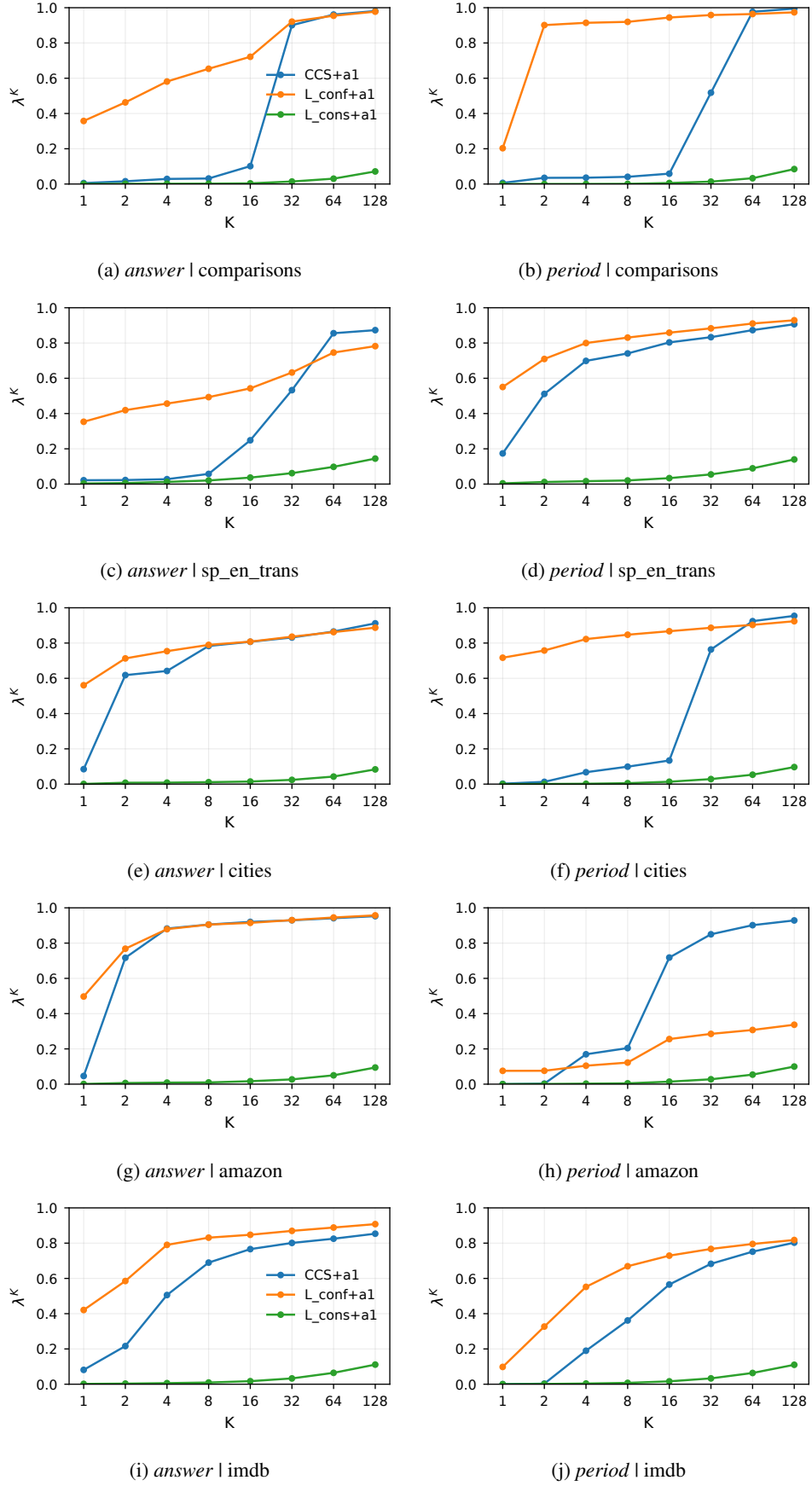


Figure 7: Effect of loss terms on probe parameter vector's similarity to top principal components. All objectives use Alteration 1 (unit-length $\hat{\theta}$).

Table 5: Accuracy of classifying with principal components of $\bar{\mathbf{X}}^+ - \bar{\mathbf{X}}$ for 9 datasets.

Dataset	token	PC1	PC2	PC3	PC4	PC5
comparisons	<i>answer</i>	100	51	41	52	56
sp_en_trans	<i>answer</i>	100	52	44	60	55
cities	<i>answer</i>	100	60	53	51	58
amazon	<i>answer</i>	96	54	50	49	54
imdb	<i>answer</i>	93	56	54	46	54
ent_bank	<i>answer</i>	85	48	59	58	56
snli	<i>answer</i>	63	61	76	64	55
copa	<i>answer</i>	51	70	44	57	53
rte	<i>answer</i>	52	65	50	61	52
comparisons	<i>period</i>	52	55	48	93	56
sp_en_trans	<i>period</i>	99	54	46	51	42
cities	<i>period</i>	51	95	65	61	57
amazon	<i>period</i>	51	92	50	49	54
imdb	<i>period</i>	91	60	51	48	52
ent_bank	<i>period</i>	89	52	55	64	56
snli	<i>period</i>	75	75	54	64	54
copa	<i>period</i>	50	49	47	58	53
rte	<i>period</i>	52	61	50	60	53

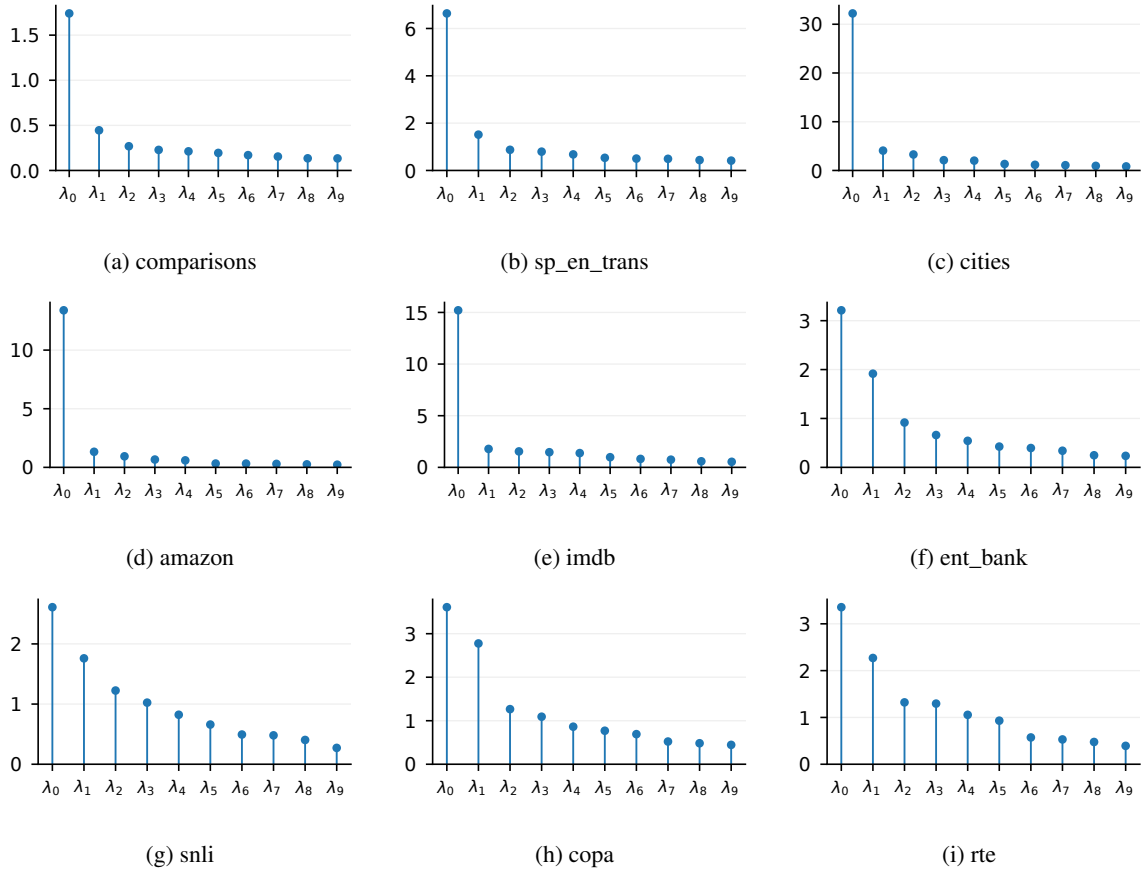


Figure 8: Top tcPCA eigenvalues for all datasets, *answer* token (Llama-2-7B).

Table 6: Strongly and weakly activating samples in COPA for the first tcPC. The answer choice corresponding to the high value is highlighted in bold.

Act. strengths	Prompt
3.72 -2.74	Consider the following example: ““The scientist conducted an experiment.”” Choice 1: She validated her theory. Choice 2: She fabricated her data. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
3.32 -3.12	Consider the following example: ““The computer crashed.”” Choice 1: I backed up my files. Choice 2: I downloaded a virus. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].
2.74 -3.46	Consider the following example: ““The man turned on the faucet.”” Choice 1: The toilet filled with water. Choice 2: Water flowed from the spout. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
3.14 -2.61	Consider the following example: ““I learned how to play the board game.”” Choice 1: My friend explained the rules to me. Choice 2: My friend got the rules wrong. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].
3.07 -2.56	Consider the following example: ““The woman tolerated her friend’s difficult behavior.”” Choice 1: The woman knew her friend was going through a hard time. Choice 2: The woman felt that her friend took advantage of her kindness. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].
2.75 -2.79	Consider the following example: ““The woman was in a bad mood.”” Choice 1: She engaged in small talk with her friend. Choice 2: She told her friend to leave her alone. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
2.61 -2.93	Consider the following example: ““I clumsily bumped into the stranger.”” Choice 1: I ran away. Choice 2: I apologized to him. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
2.55 -2.98	Consider the following example: ““The boy had trouble buttoning his shirt.”” Choice 1: He refused to wear the shirt. Choice 2: He asked his mother for help. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
3.19 -2.20	Consider the following example: ““The host served dinner to his guests.”” Choice 1: His guests were gracious. Choice 2: His guests went hungry. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
2.41 -2.96	Consider the following example: ““A man cut in front of me in the long line.”” Choice 1: I confronted him. Choice 2: I smiled at him. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
0.05 -0.06	Consider the following example: ““The water in the teapot started to boil.”” Choice 1: The teapot cooled. Choice 2: The teapot whistled. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
0.09 -0.01	Consider the following example: ““The woman received a diploma.”” Choice 1: She enrolled in college. Choice 2: She graduated from college. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].
0.08 -0.10	Consider the following example: ““The man urgently leaped out of bed.”” Choice 1: He wanted to shut off the alarm clock. Choice 2: He wanted to iron his pants before work. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].
0.11 -0.13	Consider the following example: ““The cup of tea was scalding hot.”” Choice 1: I blew on it. Choice 2: I poured it out. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
-0.02 -0.13	Consider the following example: ““The woman felt lonely.”” Choice 1: She renovated her kitchen. Choice 2: She adopted a cat. Q: Which one is more likely to be the effect, choice 1 or choice 2? choice [1/2].
0.14 0.08	Consider the following example: ““The girl’s mouth ached.”” Choice 1: She lost a tooth. Choice 2: She swallowed her gum. Q: Which one is more likely to be the cause, choice 1 or choice 2? choice [1/2].

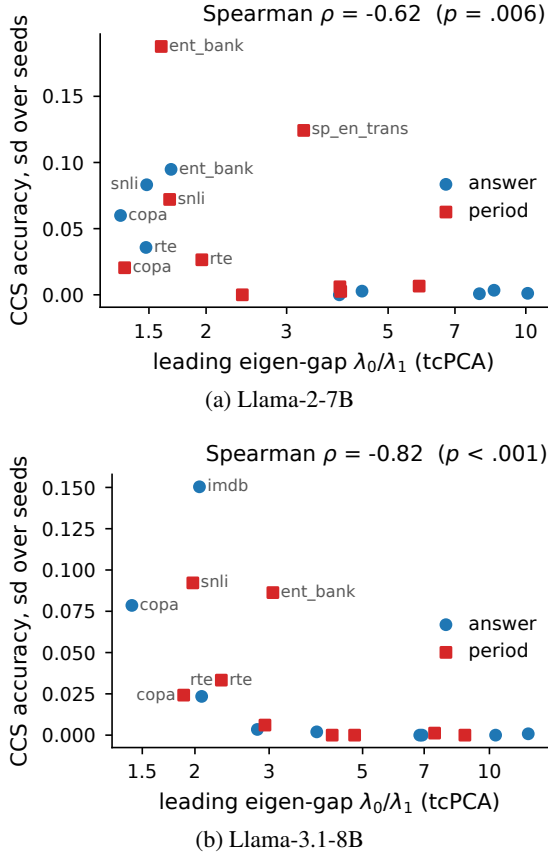


Figure 9: Spread of CCS accuracy over 30 seeds against the leading tcPCA eigen-gap, one point per dataset and token. Only datasets whose seeds vary (standard deviation of at least 0.01) are labeled; the unlabeled points all have near-identical seeds.

the plane the figure shows: nearest-centroid accuracy on held-out samples, with the centroids fitted on the training samples, in the first two components (Figure 12). For the days of the week, only layers at which the class means follow calendar order are eligible, and ties go to the layer closest to block 15.

L Full Truth-Negation Plot

A plot of the activations projected onto the first 3 tuple-contrastive Principal Components can be seen in Figure 13. The g-tcPCA views of the same data are in Figure 4; interactive versions of both are on the companion website.

M Results on Llama-3.1-8B

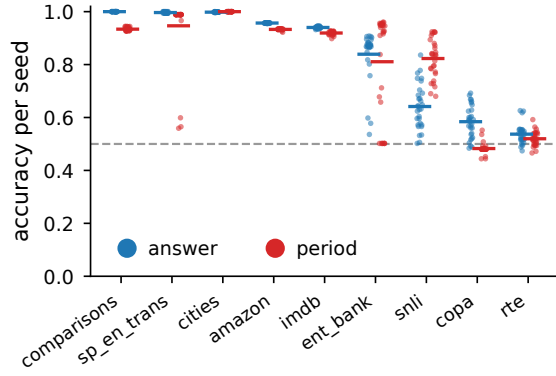
We repeated all experiments on Llama-3.1-8B with the same protocol; Tables 7, 8, 9 and 10 and Figure 14 give the results. Most findings transfer. CCS is again unreliable across seeds on the datasets with diffuse spectra, and the link between the eigen-gap

and the seed spread is stronger than on Llama-2-7B (Appendix I). tcPCA matches the median CCS probe to within one point on every dataset (Table 9). The geometric results reproduce: the four statement types again form a tetrahedron that is square from all three sides in the g-tcPCA basis, true, false and neither form a triangle, and the days lie on loops in order (Figure 6).

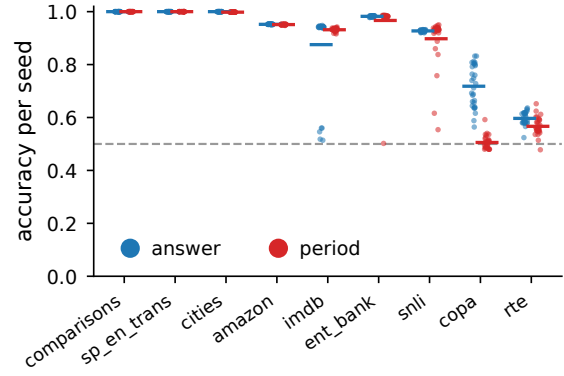
Two findings do not transfer. First, CRC-TPC does not fail on the *period* token: it matches tcPCA on comparisons, cities and amazon, and falls behind only on imdb (75% against 92%). Second, on COPA it is the first tcPC, not the second, that separates the correct from the incorrect choice: it reaches 71% (Table 10), close to the mean CCS seed (72%), whereas on Llama-2-7B the first tcPC is at chance.

Table 7: Llama-3.1-8B: CCS accuracy, mean and standard deviation over 30 seeds.

Dataset	<i>answer</i> (%)	<i>period</i> (%)
comparisons	100 ± 00	100 ± 00
sp_en_trans	100 ± 00	100 ± 00
cities	100 ± 00	100 ± 00
amazon	95 ± 00	95 ± 00
imdb	88 ± 15	93 ± 01
ent_bank	98 ± 00	97 ± 09
snli	93 ± 00	90 ± 09
copa	72 ± 08	51 ± 02
rte	60 ± 02	57 ± 03



(a) Llama-2-7B



(b) Llama-3.1-8B

Figure 10: Accuracy of every CCS seed (30 per dataset and token); dashes mark the mean.

Table 8: Llama-3.1-8B: accuracies for ablated and altered objectives.

$\mu \pm \sigma$ (%)		CCS	$\mathcal{L}_{\text{conf}}$	$\mathcal{L}_{\text{cons}}$	$\mathcal{L}_{\text{cons}}$	$\mathcal{L}_{\text{cons}}$	$\mathcal{L}_{\text{cons}}$	CCS
		-	-	-	+a1	+a2	+a1+a2	+a1+a2
Marks and Tegmark (2024)	comparisons	100±00	100±00	69±13	69±13	73±15	64±11	100±00
	sp_en_trans	100±00	96±07	66±11	65±11	72±14	65±11	100±00
	cities	100±00	98±06	67±14	67±13	70±12	65±10	100±00
Burns et al. (2023)	amazon	95±00	95±00	66±10	66±11	75±11	66±10	95±00
	imdb	88±15	54±01	58±07	59±06	65±10	61±08	90±10

Table 9: Llama-3.1-8B: tcPCA and g-tcPCA compared to CCS, *period* token.

Dataset	min	CCS med	max	CRC -TPC	tcPCA	g-tcPCA
comparisons	100	100	100	100	100	100
sp_en_trans	100	100	100	100	100	100
cities	100	100	100	100	100	100
amazon	95	95	95	95	95	95
imdb	92	93	94	75	92	92
ent_bank	50	98	99	98	97	98
snli	55	93	95	91	93	93
copa	48	50	59	50	50	50
rte	48	56	65	56	55	56

Table 10: Llama-3.1-8B: COPA, accuracy and eigenvalue of the first four tcPCs.

token		tcPC 1	tcPC 2	tcPC 3	tcPC 4
answer	acc (%)	71	66	50	53
	λ	0.31	0.22	0.18	0.16
period	acc (%)	50	56	52	59
	λ	0.19	0.10	0.09	0.07

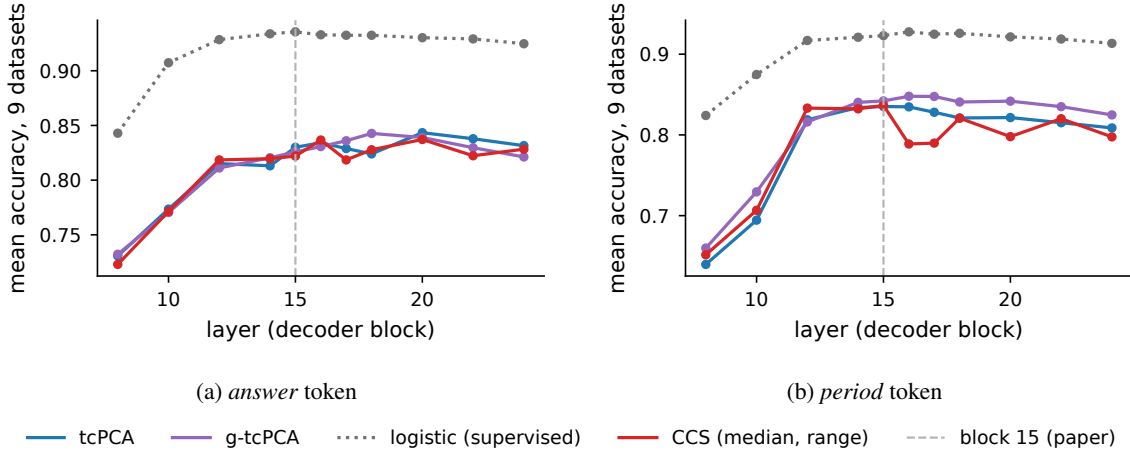


Figure 11: Accuracy against layer, averaged over the nine datasets (CCS: median over 5 seeds per dataset first), Llama-2-7B.

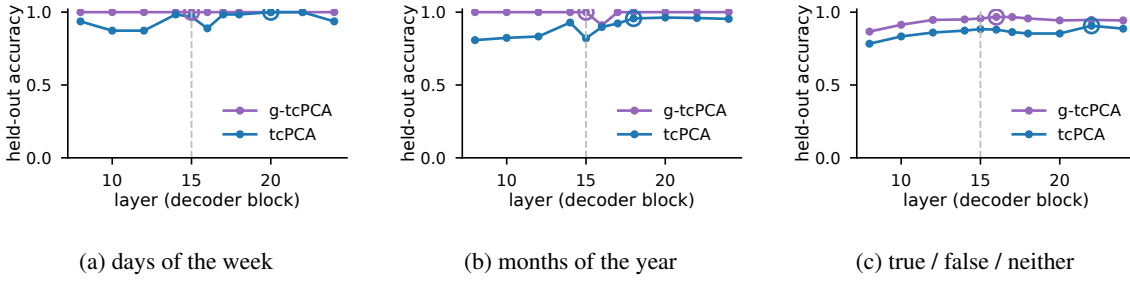


Figure 12: Choosing the layer for the non-binary figures (Llama-2-7B): held-out nearest-centroid accuracy in the plane the figure shows (the first two components), per layer, for tcPCA and g-tcPCA; circles mark the chosen layer, dashed line block 15.

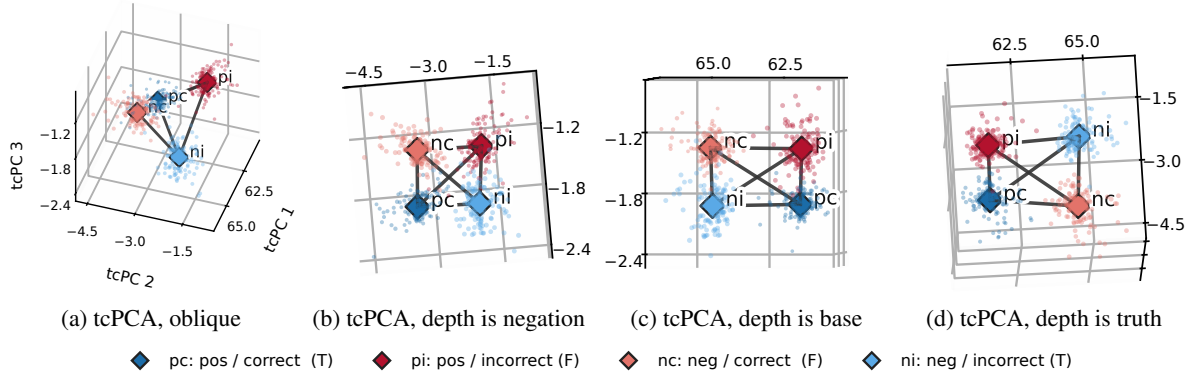


Figure 13: $\langle \mathbf{x}^{p,c}, \mathbf{x}^{p,i}, \mathbf{x}^{n,c}, \mathbf{x}^{n,i} \rangle$ on the first three components of tcPCA, Llama-2-7B (the g-tcPCA views are Figure 4). The canonical views look down the axis that collapses one contrast.

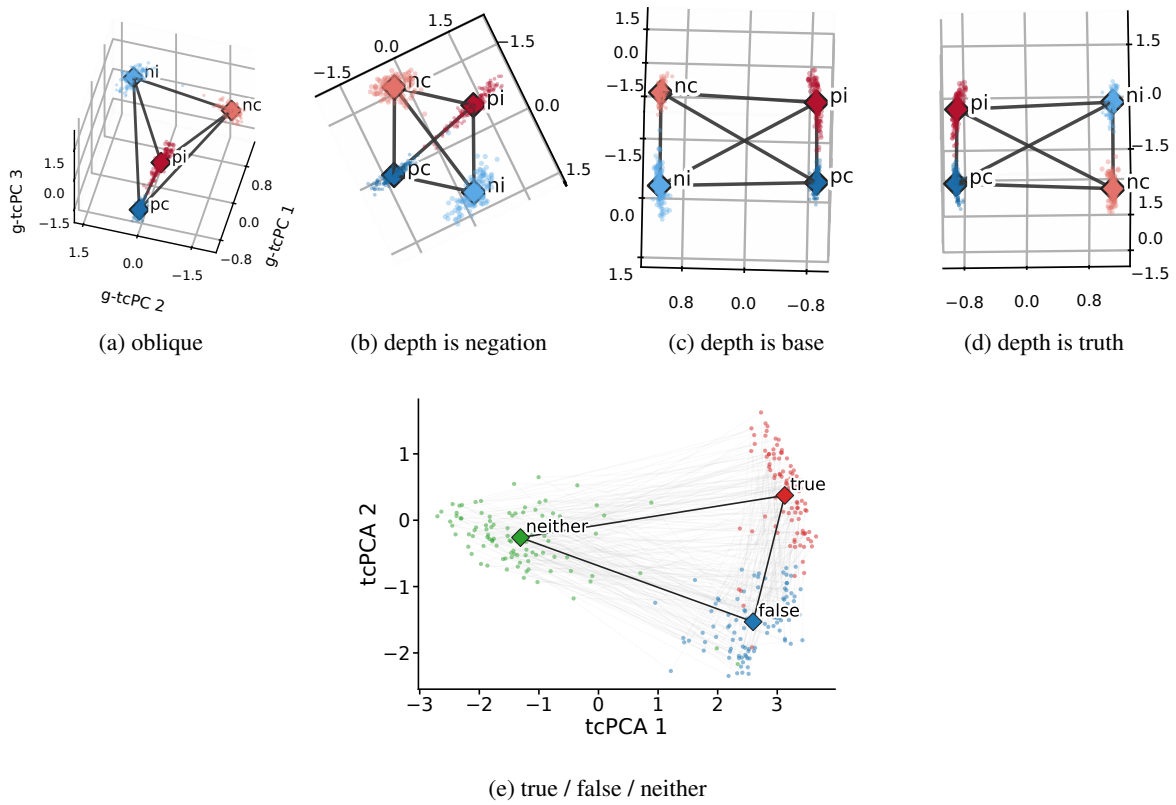


Figure 14: Llama-3.1-8B: the negation×truth tetrahedron (g-tcPCA, top) and true / false / neither (tcPCA, bottom). Days of the week: Figure 6.