

Figure 2: Difference $\log P(\text{TRUE}) - \log P(\text{FALSE})$ in LLaMA-2-13B log probabilities after patching residual stream activation in the indicated token position and layer.

despite this continuation being false. Together with the likely dataset, this will help us establish that the linear structure we observe in LLM representations is not due to LLMs linearly representing the difference between probable and improbable text.

3 Localizing truth representations via patching

Before beginning our study of LLM truth representations, we first address the question of which hidden states might contain such representations. We use simple patching experiments (Vig et al., 2020; Finlayson et al., 2021; Meng et al., 2022; Geiger et al., 2020) to localize certain hidden states for further analysis. Consider the following prompt p_F :

The city of Tokyo is in Japan. This statement is: TRUE
 The city of Hanoi is in Poland. This statement is: FALSE
 The city of Chicago is in Canada. This statement is:

Similarly, let p_T be the prompt obtained from p_F by replacing “Chicago” with “Toronto,” thereby making the final statement true. In order to localize causally implicated hidden states, we run our model M on the input p_T and cache the residual stream activations $h_{i,\ell}(p_T)$ for each token position i and layer ℓ . Then, for each i and ℓ , we run M on p_F but modify M ’s forward pass by swapping out the residual stream activation $h_{i,\ell}(p_F)$ for $h_{i,\ell}(p_T)$ (and allowing this change to affect downstream computations); for each of these intervention experiments, we record the difference in log probability between the tokens “TRUE” and “FALSE”; the larger this difference, the more causally influential the hidden state in position i and layer ℓ is on the model’s prediction.

Results for LLaMA-2-13B and the cities dataset are shown in Fig. 2; see App. B for results on more models and datasets. We see three groups of causally implicated hidden states. The final group, labeled (c), directly encodes the model’s prediction: after applying the LLM’s decoder head directly to these hidden states, the top logits belong to tokens like “true,” “True,” and “TRUE.” The first group, labeled (a), likely encodes the LLM’s representation of “Chicago” or “Toronto.”

What does group (b) encode? The position of this group—over the final token of the statement and end-of-sentence punctuation²—suggests that it encodes information pertaining to the full statement. Since the information encoded is also causally influential on the model’s decision to output “TRUE” or “FALSE,” we hypothesize that these hidden states store a

²This *summarization* behavior, in which information about clauses is encoded over clause-ending punctuation tokens, was also noted in Tigges et al. (2023). We note that the largest LLaMA model displays this summarization behavior in a more context-dependent way; see App. B.

representation of the statement’s truth. In the remainder of this paper, we systematically study these hidden states.

4 Visualizing LLM representations of true/false datasets

We begin our investigation with a simple technique: visualizing LLMs representations of our datasets using principal component analysis (PCA). Guided by the results of §3, we present here visualizations of the *most downstream* hidden state in group (b); for example, for LLaMA-2-13B, we use the layer 15 residual stream activation over the end-of-sentence punctuation token.³ Unlike in §3, we do not prepend the statements with a few-shot prompt (so our models are not “primed” to consider the truth value of our statements). For each dataset, we also center the activations by subtracting off their mean.

When visualizing LLaMA-2-13B and 70B representations of our curated datasets – datasets constructed to have little variation with respect to non-truth features, such as sentence structure or subject matter – **we see clear linear structure** (Fig. 1), with true statements separating from false ones in the top two principal components (PCs). As explored in App. C, this structure emerges rapidly in early-middle layers and emerges later for datasets of more structurally complex statements (e.g. conjunctive statements).

To what extent does this visually-apparent linear structure align between different datasets? Our visualizations indicate a nuanced answer: **the axes of separation for various true/false datasets align often, but not always**. For instance, Fig. 3(a) shows the first PC of *cities* also separating true/false statements from other datasets, including diverse uncured datasets. On the other hand, Fig. 3(c) shows stark failures of alignment, with the axes of separation for datasets and statements and their negations being approximately orthogonal.

These cases of misalignment have an interesting relationship to scale. Fig. 3(b) shows *larger_than* and *smaller_than* separating along *antipodal* directions in LLaMA-2-13B, but along a common direction in LLaMA-2-70B. App. C depicts a similar phenomenon occurring over the layers of LLaMA-2-13B: in early layers, *cities* and *neg_cities* separate antipodally, before rotating to lie orthogonally (as in Fig. 3(c)), and finally aligning in later layers.

4.1 Discussion

Overall, these visualizations suggest that as LLMs scale (and perhaps, also as a fixed LLM progresses through its forward pass), they hierarchically develop and linearly represent increasingly general abstractions. Small models represent surface-level characteristics of their inputs, and large models linearly represent more abstract concepts, potentially including notions like “truth” that capture shared properties of topically and structurally diverse inputs. In middle regimes, we may find linear representation of concepts at intermediate levels of abstraction, for example, “accurate factual recall” or “close association” (in the sense that “Beijing” and “China” are closely associated).

To explore these intermediate regimes more deeply, suppose that \mathcal{D} and \mathcal{D}' are true/false datasets, f^+ is a linearly-represented feature which correlates with truth on both \mathcal{D} and \mathcal{D}' , and f^- is a feature which correlates with truth on \mathcal{D} but has a negative correlation with truth on \mathcal{D}' . If f^+ is very *salient* (i.e. the datasets’ have large variance along the f -direction) and f^- is not, then we expect PCA visualizations of $\mathcal{D} \cup \mathcal{D}'$ to show joint separation along f^+ . If f^- is very salient but f^+ is not, we expect antipodal separation along f^- , as in Fig. 3(b, center). And if both f^+ and f^- are salient, we expect visualizations like Fig. 3(c).

To give an example, suppose that $\mathcal{D} = \text{cities}$, $\mathcal{D}' = \text{neg_cities}$, $f^+ = \text{“truth”}$, and $f^- = \text{“close association”}$. Then we might expect f^- to correlate with truth positively on \mathcal{D} and negatively on \mathcal{D}' . If so, we would expect training linear probes on $\mathcal{D} \cup \mathcal{D}'$ to result in

³Our qualitative results are insensitive to choice of layer among early-middle to late-middle layers. On the other hand, when using representations over the final token in the statement (instead of the punctuation token), we sometimes see that the top PCs instead capture variation in the token itself (e.g. clusters for statements ending in “China” regardless of their truth value).

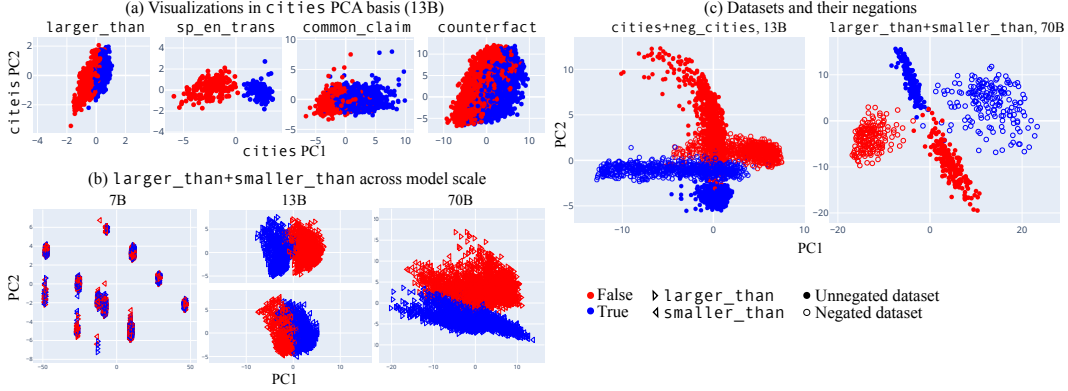


Figure 3: (a) Projections of LLaMA-2-13B onto the top 2 PCs of cities. (b) PCA visualizations of larger_than+smaller_than. For LLaMA-2-7B (left), we see statements cluster according to surface-level characteristics, e.g. presence of the token “eighty.” For LLaMA-2-13B, we see that larger_than (center, top) and smaller_than (center, bottom) separate along opposite directions. (c) PCA visualizations of datasets and their negations. Unlike in other visualizations, we use layer 12 for cities+neg_cities; see App. C for an exploration of this misalignment emerging and resolving across layers.

improved generalization, despite \mathcal{D}' consisting of the same statements as \mathcal{D} , but with the word “not” inserted. We investigate this in §5.

5 Probing and generalization experiments

In this section we train probes on datasets of true/false statements and test their generalization to other datasets. But first we discuss a deficiency of logistic regression and propose a simple, optimization-free alternative: **mass-mean probing**. Concretely, mass-mean probes use a difference-in-means direction, but—when the covariance matrix of the classification data is known (e.g. when working with IID data)—apply a correction intended to mitigate interference from non-orthogonal features. We will see that mass-mean probes are similarly accurate to probes trained with other techniques (including on out-of-distribution data) while being more causally implicated in model outputs.

5.1 Challenges with logistic regression, and mass-mean probing

A common technique in interpretability research for identifying feature directions is training linear probes with logistic regression (LR; [Alain & Bengio, 2018](#)). In some cases, however, the direction identified by LR can fail to reflect an intuitive best guess for the feature direction, even in the absence of confounding features. Consider the following scenario, illustrated in Fig. 4 with hypothetical data:

- Truth is represented linearly along a direction θ_t .
- Another feature f is represented linearly along a direction θ_f not orthogonal to θ_t .⁴
- The statements in our dataset have some variation with respect to feature f , independent of their truth value.

We would like to identify the direction θ_t , but LR fails to do so. Assuming for simplicity linearly separable data, LR instead converges to the maximum margin separator [Soudry et al. \(2018\)](#) (the dashed magenta line in Fig. 4). Intuitively, LR treats the small projection of θ_f onto θ_t as significant, and adjusts the probe direction to have less “interference” ([Elhage et al., 2022](#)) from θ_f .

⁴The *superposition hypothesis* of [Elhage et al. \(2022\)](#), suggests this may be typical in deep networks.