What are They Thinking? Delineation, Probing and Tracking of Concepts in LLMs

Mohamed Abdelwahab[§] **Michelle Yu Collins** **Sihan Chen** **Yi Cheng Zhao**
Zafarullah Mahmood **Jiading Zhu** **Soliman Ali** **Jonathan Rose**
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto

Abstract

As the influence of LLMs expands, it is imperative to gain insight into their decisions. One way to do that is to develop probes that detect the presence or absence of a broad set of concepts within the embeddings computed in an LLM - which is what we might say a model is "thinking" about. Such probes should be low-cost and easily applicable to any LLM, so that monitoring for many concepts is possible during normal operation.

In this paper, we take the first steps towards developing the capability of creating many such probes by defining and executing examples of the key tasks needed: first, the careful *delineation* of a concept through the creation of a dataset with the concept both present and then absent. Then, the training and testing of a set of linear probes to detect the concept on any layer of an LLM, including an exploration of the complexity of the probe needed. Finally, we show that such probes can track concepts across larger contexts. This is done with four separate concepts and three different LLMs. When this process is scaled to many more concepts, it will create the ability to easily monitor new models.

1 Introduction

Large Language Models (LLMs) appear to function as *concept machines*, in that they infer implied concepts (at multiple levels of abstraction) from their input that are then the driving force behind the generated output. Indeed, it is now well understood that many kinds of concepts can be detected in internal embeddings within LLMs using linear probes (Conneau et al., 2018; Hupkes and Zuidema, 2018). These concepts include Parts-of-Speech tags, verb tense (Liu et al., 2019; Ravishankar et al., 2019; Arps et al., 2022; Hewitt and Manning, 2019; Durrani et al., 2020; Tenney et al., 2019; Kim et al., 2019) as well as time, physical location (Gurnee and Tegmark, 2024), and truth (Burns et al., 2024). There are many other important concepts whose presence or absence needs to be identified, and so the goal in this work is to prototype the steps needed to create a broad set of probes that could be used to monitor an LLM.

The first step is to select a set of concepts that are needed for the downstream task of monitoring. In these first steps, we select the general goal of human activities, and (somewhat arbitrarily) choose the concepts of ambition, investigation, democracy, and envy. The next step is to create a working definition of the term *concept* itself, based on the prototype theory (Ren and Wei, 2019) of concepts: a concept is an entity characterized by a set of features used to determine membership within it. Finally, we use the term 'delineation' of a concept to mean the method by which we create labeled examples considered to embody the concept and those that do not.

To illustrate the use of these probes for monitoring, we show how they can track the waxing and waning of concepts in an LLM's embeddings as more words are added to the input context. For example, Figure 1 illustrates this using a probe trained to detect **ambition** in the `Llama-3-8B` model. The probe is applied to the final embedding of an expanding input sequence, taken from the 13^{th} transformer layer. The X-axis gives the input tokens (as complete words), and the Y-axis gives the probe's sigmoid output computed on the embedding *after* each new word is added to the sequence. We take a probe output above 0.5 to indicate the presence of the concept, while an output below 0.5 indicates its absence. The colored shading in the figure indicates the label of the *entire* sentence: green for the presence of ambition and red for its absence. The sentences in the figure are drawn from the middle of a three-paragraph story, and it is instructive to read them and compare them with the probe's out-

[§]Corresponding author:
mo.abdelwahab@mail.utoronto.ca

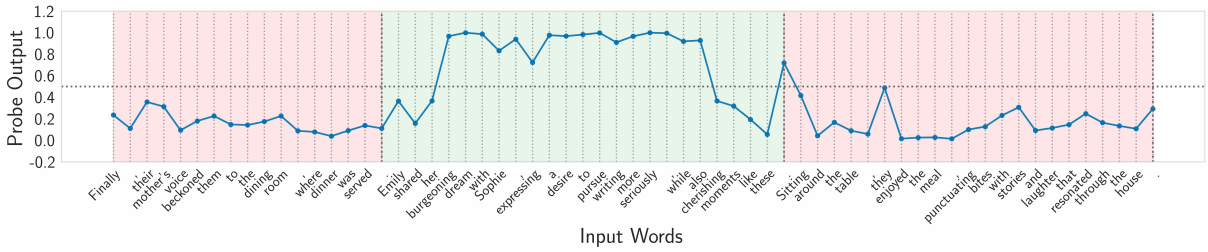

Figure 1: **Ambition** probe output for every word in an expanding context in layer 13 of `Llama-3-8B`

puts. We observe that the probe output rises above 0.5 when the newly added span of words implies ambition, and falls below 0.5 once the continuing context no longer does.

Previous work has explored the detection of concepts in LLMs, including the use of Sparse Autoencoders (SAEs) (Cunningham et al., 2023; Bricken et al., 2023; Lieberum et al., 2024; Rajamanoharan et al., 2024; Templeton et al., 2024) to detect a large number of concepts in an LLM. This is done using unsupervised training of an SAE that is tasked with reconstructing an internal embedding from a large (enforced) sparse representation. This approach creates many concept detectors, but does not have anything equivalent to our stage of *delineation* where the specific concept is chosen and all others are excluded. SAEs do not permit control over the concepts that can be detected, which is very problematic, especially in matters of trust. In addition, training is very computationally expensive and must be repeated for every new model. By contrast, our approach builds a delineating dataset only once per concept, enabling inexpensive probe training on *any model* thereafter.

Another approach to concept detection, proposed in (Zou et al., 2023), allows direct specification of a concept by explicitly instructing the LLM to identify a named concept. However, this makes it unsuitable for continuous monitoring, as it requires a separate LLM invocation for detection.

In this work, we create probes for four implied concepts – **ambition**, **investigation**, **democracy**, and **envy** – by constructing a dataset with validated presence/absence labels for each. These labels serve as ground truth for detecting the concepts within the LLM embeddings. This method requires significantly less training than SAEs and allows direct specification of the concepts to study.

The primary contributions of this paper are: (1) We propose a semi-automatic LLM-based method for delineating a concept through the creation of a binary dataset of textual examples for a specified concept. The method seeks to inhibit patterns that may unintentionally give away the labels; (2) We illustrate the utility of the delineation method by training linear probes on the binary datasets and show that these achieve good accuracy. (3) We show how the size of the probes can be constrained to fewer than 80 parameters while still achieving good accuracy. (4) We show the probes used to observe concepts waxing and waning in the model embeddings as words are added to the input context, illustrating their use as low-cost continuous monitors; (5) We provide the created datasets for others to use. Since each dataset can be reused on any LLM to build probes, we believe this motivates a larger effort to select and delineate many such reusable concept datasets to advance research in LLM explainability.

2 Background and Related Work

2.1 Concepts in LLMs

Several works explore the existence of concepts in LLMs. Shani et al. (2023) and Liao et al. (2023) investigate LLMs' knowledge of concept hierarchies. Shani et al. (2023) use zero-shot prompting (Brown et al., 2020), directly asking the model to generate an answer on whether one concept is within the broader category of another. In contrast, Liao et al. (2023) present the model with statements expressing conceptual relationships and probe its embeddings to assess the validity of those relations.

Zou et al. (2023) explore concepts in LLMs by extracting embeddings after prompting the model to identify a specific concept in an input example. They apply PCA (Pearson, 1901; Hotelling, 1933) to these embeddings to derive a 'concept vector.' To detect the concept in new inputs, they use the same prompt to extract an embedding and measure its alignment with the concept vector using dot product. However, this approach is unsuitable for continuous concept monitoring, since detection

requires a separate LLM invocation.

Cunningham et al. (2023); Bricken et al. (2023); Lieberum et al. (2024); Rajamanoharan et al. (2024); Templeton et al. (2024) use SAEs to disentangle LLM embeddings into many dimensions corresponding to individual concepts. SAEs undergo unsupervised training, so concepts are identified post hoc using an automated LLM-based method (Bills et al., 2023). As such, the extracted concepts cannot be specified in advance, offering no guarantee of cross-model consistency, and SAEs demand large-scale training (>1B examples per SAE). There is also no guarantee of disentanglement. As a result, SAEs are not suitable for dedicated investigations into LLMs' inference of concepts.

Our work proposes a probing-based approach to perform concept detection in LLMs using datasets constructed for specific concepts. This method enables exploration of these concepts across models, requires far less training than SAEs, and minimizes the risk of cueing the model toward the target concept. It also enables low-cost monitoring of how concepts wax and wane in a model's embeddings as the context expands during generation.

2.2 Linear LLM Probes

LLM probes have been used to study the properties an LLM has acquired during training by using a separate model that makes a prediction given an LLM embedding. The probe is usually a classifier model that is trained to detect a specific property. If the probe achieves a reasonable accuracy, it suggests that the property is encoded within the LLM.

Probes were first applied to early transformer models, such as BERT (Devlin et al., 2019), to explore whether they encoded linguistic properties such as Parts-of-Speech tags and main verb tense (Liu et al., 2019; Ravishankar et al., 2019; Arps et al., 2022; Hewitt and Manning, 2019; Durrani et al., 2020; Tenney et al., 2019; Kim et al., 2019). More recently, probes have been used to explore the encoding of time, space, and truth within LLMs (Gurnee and Tegmark, 2024; Burns et al., 2024).

3 Detection and Tracking of Concepts within LLMs

We use linear probe classifiers to detect concepts in LLMs, and to illustrate how these concepts wax and wane in a model's embeddings as the input context expands. In this way, they can be used for monitoring the LLMs for "thinking" about a specific concept.

3.1 Inference of Concepts in LLMs

The training of a probe requires a binary dataset of textual examples for the concept. The concept should be *present* in positive examples and *absent* in negative ones. These examples are fed into an LLM to extract embeddings, which are then used to train and evaluate a probing classifier.

3.1.1 Concept Dataset Creation

When creating a binary dataset for a concept, the goal is to ensure that the positive and negative examples differ solely in the presence or absence of the concept, without unintentionally including any patterns that could "give away" the label (Geirhos et al., 2019; Xiao et al., 2021; Wang and Wang, 2024). This is achieved by using textual example templates that were created independently of any particular concept. We then generate a pair of examples following the linguistic structure of the template, where the concept is present in one example and absent in the other. Table 1 shows an example template and a positive-negative example pair generated from it; additional examples are provided in Appendix A.

We obtain templates from a dataset based on Project Gutenberg (Faysse, n.d.; Project Gutenberg, n.d.), a free eBook library. Paragraphs from English books are split into sentences, and templates are formed by randomly selecting one to three consecutive sentences. We remove templates with incomplete sentences, misplaced words or numbers, or those not centered on human subjects, since we focus on human-related concepts in this work. A prompted LLM[1] is used for filtering. The prompt used, as well as all other prompts employed throughout this paper, are provided in Appendix B. We created a set of 30,000 templates, which can be reused to create any specific concept dataset.

To create a dataset for a specific concept, the concept is first defined as having specific features, following the definition of *concept* in Section 1. This definition is then provided to two instances of the same LLM: one prompted to generate positive (concept present) examples, and the other to generate negative (absent) examples. The prompts instruct the model to mimic the structure of the given example templates while altering the semantics to reflect the presence or absence of the concept.

[1]We use `gpt-4o-2024-08-06` (OpenAI, 2024) for all prompted generators and classifiers.

Example template	"I want the girls to understand this," said Miss Anstice with decision.
Positive Example (Concept Present)	"I aim to reach a million followers this year," announced Jake confidently.
Negative Example (Concept Absent)	"I need the followers to see this," posted Sarah with certainty.

Table 1: Example-pair created for **Ambition** dataset using an example template

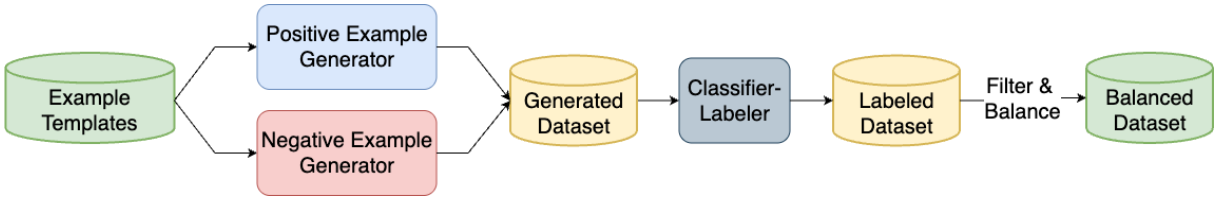

Figure 2: Creation of a concept dataset while limiting label leakage

We diversify the generated examples by instructing the LLM to generate the examples within a specific context that is changed every five examples. The contexts include *workplace*, *academia*, *sports*, *entrepreneurship*, *politics*, *arts*, *music*, *community*, *science*, *technology*, and *social media*. Details on the prompt refinement are provided in Appendix B.

Because the generated examples may not reliably match their intended labels, and manual verification at scale is impractical, a prompted LLM-based classifier is used to re-label all examples for greater reliability. The classifier is instructed to follow the concept definition, and the prompt is iteratively refined until it exceeds 90% accuracy on at least 360 manually labeled examples. During manual labeling, human annotators also follow the concept definition. Appendix B details the labeling process and inter-rater reliability. Figure 2 illustrates the generation and labeling workflow.

To ensure a balanced dataset, we filter it to retain only example pairs with opposite classifier-assigned labels. This prevents the probe from exploiting structural similarities between examples that share the same label.

Four datasets were created using this method for the concepts of **ambition** (with 11,854 examples), **investigation** (8,296), **democracy** (10,270), and **envy** (15,350). The specific definitions of these concepts are provided in Appendix B. These concepts were selected because they span various concept categories, as discussed in Appendix C. A 70/10/20 train/validation/test split is used for probe training and evaluation.

We recognize that one might argue that an LLM-prompted classifier, which labels the presence or absence of a concept, could directly be used for concept monitoring. This is not practical for two reasons: first, it is a computationally expensive monitor that would not be practical to use for many monitors. Second, the prompted classifier would only be inferring the 'thoughts' of the target model from its language, and not the actual concepts the target was traversing, such as when the model is trying to mislead (Greenblatt et al., 2024).

3.1.2 Probe Training for a Concept

To train a probe for a specific concept, each example in the dataset is input into the LLM. The embeddings are extracted from every layer of the model to explore where the concept may appear. To clarify the extraction process of the embeddings, we formalize it as follows:

A textual example is tokenized into a sequence $X = \{x_1, x_2, \ldots, x_N\}$, where x_i denotes the i-th token and N is the total number of tokens. Each token is mapped to an embedding $e_i^0 \in \mathbb{R}^{d_{model}}$ via the embedding layer, where d_{model} is the LLM's embedding size. These embeddings are then processed through the model's transformer layers, with layer ℓ producing the set $E^\ell = \{e_1^\ell, \ldots, e_N^\ell\}$.

We train the probe using one of two vectors for each layer ℓ: (1) the N^{th} embedding e_N^ℓ, which encodes information for the entire context up till token N, or (2) the average of all embeddings in E^ℓ. We refer to either as the *representative embedding*.

After obtaining a representative embedding for each example in the labeled dataset, it is assigned the label of the corresponding input text. The training split of this (embedding, label) dataset is used to train a linear probe (Conneau et al., 2018; Hupkes and Zuidema, 2018)–a binary linear classifier–to predict the presence or absence of the concept. The

probe, trained via gradient descent, is evaluated on the test set. It contains d_{model} parameters.

A common criticism of probing is that the probe may learn the target feature *itself* rather than detect its existence in the model's embeddings (Hewitt and Liang, 2019; Kunz and Kuhlmann, 2020; Zhu and Rudzicz, 2020). To test for this, the literature suggests control tasks such as training probes on randomized data. Here, we randomize the training and validation sets–either the embeddings[2] or the labels–and then evaluate the probe on the original test set. A sharp accuracy drop with randomized embeddings indicates that the probe depends on information encoded in the embeddings. Likewise, a drop with randomized labels suggests that the probe's performance relies on a meaningful mapping between embeddings and labels, rather than a superficial one.

Another control task involves reducing the probe's learning capacity by reducing its number of parameters. To do so, we apply PCA to reduce the dimensionality of the embeddings(from 1 up to d_{model})[3], thereby reducing the parameters of the linear probe, which are equal to the embedding size. The probe's accuracy is evaluated at each dimensionality level; if it remains largely unchanged despite substantial compression, this indicates that the embeddings encode the feature, not the probe.

3.2 Waxing and Waning of Concepts

In Section 4.1.2 below, we confirm that LLMs can infer concepts. We were also interested to see if the same trained probes could be used to examine whether and how a concept waxes and wanes in the model embeddings as the input context grows. This analysis requires longer-text datasets to which the probes are applied. These will be referred to as the *Story Datasets*.

3.2.1 Story Dataset Creation

For each concept, a dataset of three-paragraph stories is constructed. Each paragraph contains 10 sentences, and each pair of paragraphs is connected by a transition sentence, totaling 32 sentences per story. The studied concept is present only in the transition sentences and is absent elsewhere. The stories are generated using a prompted LLM.

The generator prompt defines the target concept and instructs the LLM to generate a story in which the concept is initially absent. It then instructs the model to insert transition sentences between the paragraphs, each having the concept present. The prompt also emphasizes maintaining semantic coherence throughout. Appendix D provides further details on the generation process.

The prompted classifier, described in Section 3.1.1, is used to label each sentence in the story individually, retaining only stories where the concept appears only in the transitions. A total of 50 stories were created for each target concept.

3.2.2 Probing for Waxing and Waning

Each story is input to the LLM and, for each word, we obtain a representative embedding which encodes the input context *up to and including* that word. Let the story consist of words (including punctuation) $W = \{w_1, \ldots, w_S\}$, tokenized into *subword* tokens $X = \{x_1, \ldots, x_{S'}\}$, where $S' \geq S$. At each layer ℓ, the LLM produces embeddings $E^\ell = \{e^\ell_1, \ldots, e^\ell_{S'}\}$. The *representative embedding* for word w_i will either be the embedding of its final subword token, $e^\ell_{i'}$, or the cumulative mean of all embeddings up to $e^\ell_{i'}$.

The representative embeddings in each layer are classified by the corresponding probe for the target concept. This yields a probe output vector $P = \{p_1, \ldots, p_S\}$, where each $p_i \in [0, 1]$ indicates the concept's presence (above 0.5) or absence (below 0.5) in the embedding that encodes the context up to and including word w_i. We can observe these word-level outputs and compare them to the sentence-level labels of the story, as shown in Figure 1. This allows us to assess how the LLM's encoding of the concept evolves as the story unfolds, and whether the probe output aligns with the sentence labels.

4 Experiments and Results

4.1 Inference of Implied Concepts

4.1.1 Experimental Setup

To explore concept inference in LLMs, we tested seven open-source models from three model families: `Llama-3-8B` (Grattafiori et al., 2024), `Gemma-2` (`2B`, `9B`) (Riviere et al., 2024), and `Qwen2.5` (`0.5B`, `1.5B`, `3B`, `7B`) (Yang et al., 2025). All models were accessed through the Hugging Face Transformers library (Wolf et al., 2020). The details for these models are given in Appendix E.

[2]We randomize embeddings by randomizing the input tokens to the LLM.

[3]Using principal components derived from the embeddings of the example templates.

The concepts under test are **ambition**, **investigation**, **democracy**, and **envy**, which are bolded throughout to avoid ambiguity. Datasets were created for each, as described in Section 3.1.1.

These datasets were used to produce representative embeddings[4] (described in Section 3.1.2) which were then used to train 5 probes (with different seeds) per layer, averaging their results. For the representative embedding, we experimented with both the N^{th} embedding and the mean of all embeddings on a given layer. Appendix F describes the probes' hyperparameter settings.

We primarily present results for the concept of **ambition**, with similar results for the other concepts provided in Appendix G. Since all LLMs show similar patterns, we focus on `Llama-3-8B`, reporting results for other models in Appendix G, unless otherwise noted. `Llama-3-8B` has a d_{model} size of 4,096 and 32 transformer layers.

4.1.2 Results and Discussion

Figure 3 shows a plot of the probe accuracy for each concept under test across all layers of `Llama-3-8B`. Here, the probes were trained and tested on the N^{th} embedding on each layer. The Y-axis is broken at the bottom to include all data points while maintaining the visibility of trends. All the results are well above 50% (except for the embedding layer, which is close to 50%) providing evidence that the LLM infers these concepts. We had expected that probing the embedding layer (layer 0) would be unsuccessful, as the probe would only see the uncontextualized embedding of the one final token, which is clearly not sufficient to encode a concept.

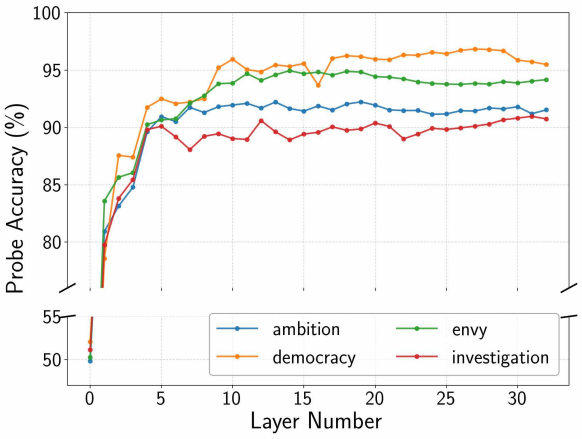

Figure 3: Concept probe accuracy across layers in `Llama-3-8B` for all 4 concepts

[4]Using a single Nvidia A100 GPU with 40GB VRAM

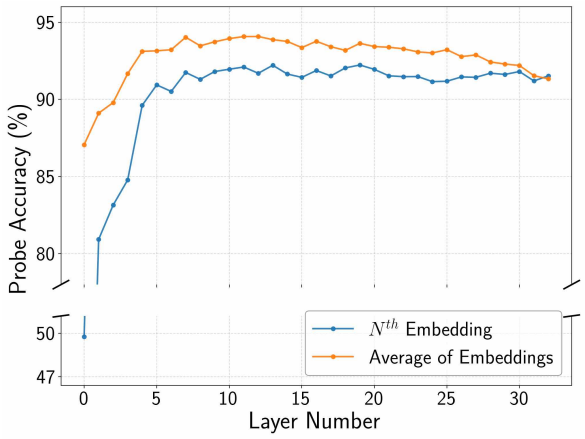

Figure 4: **Ambition** probe accuracy for `Llama-3-8B` using average and N^{th} embeddings vs. layer

Figure 4 plots the probe accuracy versus layer number for **ambition** in `Llama-3-8B`, comparing the two types of representative embeddings described in Section 3.1.2: the full-layer average embedding and the N^{th} embedding. Interestingly, the average embedding outperforms the N^{th} embedding across most layers, with the performance gap narrowing in the deeper layers.

It is interesting to observe that the average embedding from the embedding layer (layer 0) achieves 87% accuracy. These embeddings are produced during the language model's training but are typically regarded as *uncontextualized*, in contrast to the *contextualized* outputs of transformer layers. It seems that concepts can be detected in LLMs with simple contextualization—averaging—much like in a bag-of-words model. While this works for small contexts of a few sentences, we show in Section 4.2.2 that it does not work for longer contexts, as intuition suggests, and so the model computation is necessary to extract the concept.

To test whether the probe itself learns the concept, we apply the control tasks described in Section 3.1.2. We first explore probe size reduction in Figure 5 which plots **ambition** probe accuracy versus the number of probe parameters. The probe is applied to the N^{th} embedding in three layers of `Llama-3-8B`. Even with only 40 probe parameters, the accuracy is at least 75%, well above random guessing, and exhibits diminishing returns toward 100 parameters, well below the maximum of 4,096 for `Llama-3-8B`. Similar trends hold for other models of different sizes, summarized in Table 2. Accuracy drops by roughly 15% with 20 parameters and 10% with 40, while performance gains diminish

beyond 80 parameters.

In the second control task, we randomized either the embeddings or the labels and evaluated the resulting probe accuracy. In both cases, accuracy dropped to around 50% for all layers in all LLMs (Appendix G). This suggests that the probe's performance depends both on the information encoded in the embeddings and on having a meaningful, rather than superficial, mapping between embeddings and labels. We believe that this demonstrates that the probes are not learning the concepts themselves.

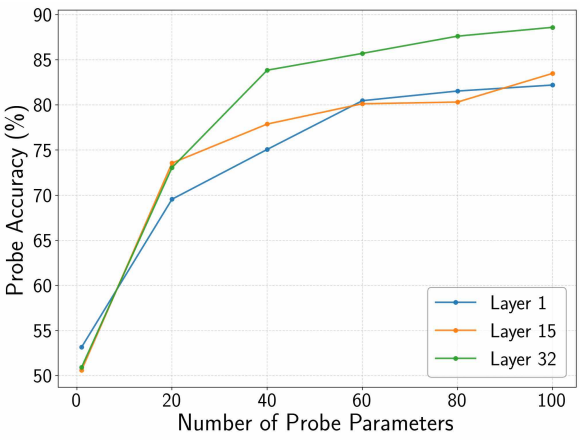

Figure 5: **Ambition** probe accuracy for `Llama-3-8B` as a function of probe size

Probed LLM	Probed Layer	# Probe parameters 20	40	80	max
`Llama-3-8B`	1	70	75	82	81
	15	74	78	80	91
	32	73	84	88	92
`Gemma-2-2B`	1	63	73	81	84
	15	77	80	83	91
	26	74	82	86	90
`Qwen2.5-0.5B`	1	65	67	74	69
	15	76	78	82	89
	24	75	80	85	89

- All results are in percentage (%).
- "max" denotes 4,096 for `Llama-3-8B`, 2,304 for `Gemma-2-2B`, and 896 for `Qwen2.5-0.5B`.
- standard deviation for each result $\leq 1\%$.

Table 2: **Ambition** probe accuracy across model families, sizes, layers, and probe sizes

4.2 Tracking Concepts across LLM Context

Our secondary goal is to investigate how an encoded concept evolves as the input context expands. To do so, we use the same probes[5] that were trained for the concepts studied in Section 4.1.

4.2.1 Experimental Setup

A set of 50 stories, each consisting of 32 sentences, was created for each concept, as described in Section 3.2.1. Recall that the target concept appears only in the two transition sentences that join three paragraphs. Our goal is to determine whether the probe can detect the waxing and waning of the concept throughout these stories.

We evaluated the seven LLMs from Section 4.1.1 using two representative embeddings for each word: the final sub-word token embedding, and the cumulative mean embedding, as detailed in Section 3.2.2. The former was paired with the concept-specific probes trained on the N^{th} embedding, while the latter was used with probes trained on the mean embedding, both discussed in Section 3.1.2.

To obtain an aggregate view of the probe's word-level behavior (and, by extension, the model embeddings' behavior) across all 50 stories, we average its sigmoid outputs for each word index across stories. Before averaging, sentences are aligned by position and padded [6] to equal length to ensure that word positions align across stories. This process is detailed in Appendix H.

As in Section 4.1, all LLMs showed similar trends across the investigated concepts. Thus, we present results for `Llama-3-8b` on **ambition**, deferring the rest to Appendix H.

4.2.2 Results and Discussion

Figure 6 shows the aggregate **Ambition** probe output across word indices, using both final sub-word token embeddings and cumulative mean embeddings from layer 13 of `Llama-3-8B`. Vertical dotted lines mark sentence boundaries, with the uneven spacing between markers reflecting different sentence lengths. Green and red backgrounds denote concept presence and absence, respectively. A 10-word moving average is applied for smoothing.

For the final sub-word token embeddings, the aggregate outputs across words in each sentence show a clear trend: they rise and surpass the 0.5 classification threshold when the concept is present, and fall below 0.5 when it is absent. The higher variation at the start of sentences is expected, as the full semantic meaning is less clear earlier on, *and*

[5] We use a single trained probe per model layer per concept, rather than the five probes used in Section 4.1.

[6] Padded positions were excluded from the computation.

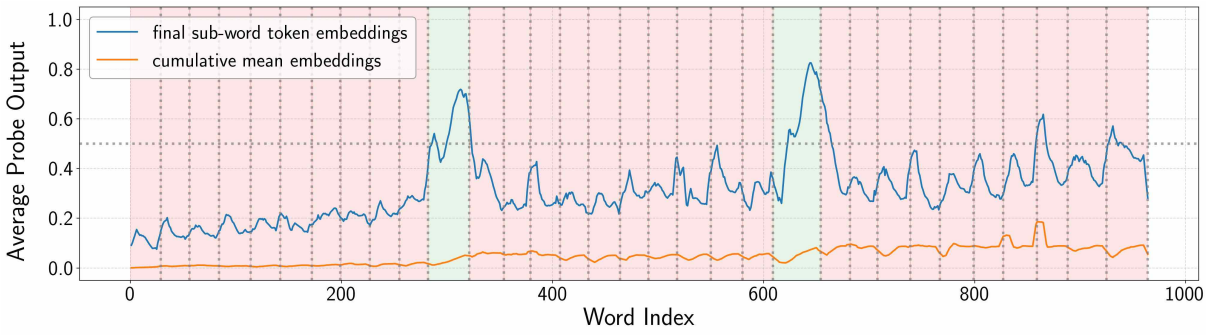

Figure 6: **Ambition** probe outputs versus word index, averaged across 50 stories in `Llama-3-8B`

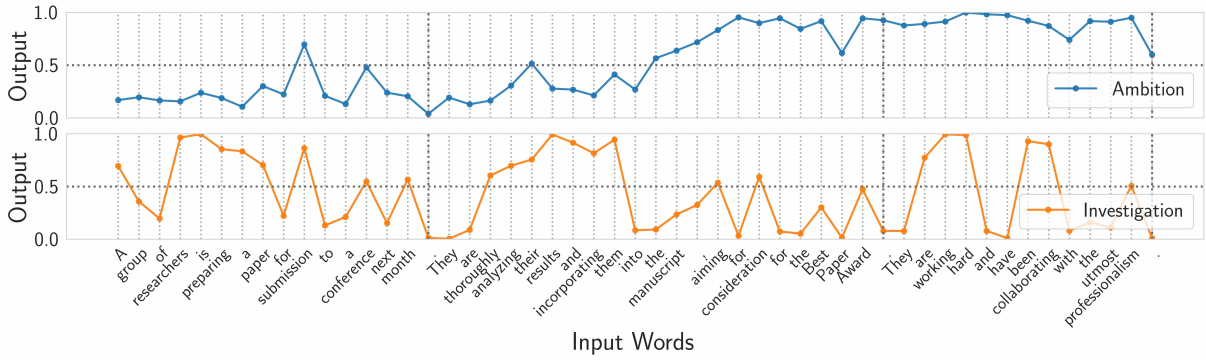

Figure 7: **Ambition** & **Investigation** probe outputs across words using final sub-word token embeddings in `Llama-3-8B`

due to the padding needed to align these results. This pattern suggests that the LLM embeddings capture changes in concept presence, which align with those in the input context, even across long contexts. We illustrate how this behavior changes across LLM layers in Appendix H.

In contrast, cumulative mean embeddings lose the waxing and waning pattern. Averaging over many tokens, most unrelated to the concept, dilutes its presence in sentences where it does exist, obscuring changes in concept presence. This supports the view that concept inference in LLMs goes beyond a simple bag-of-words representation, contrary to what one might conclude from Section 4.1.2.

This method of tracking concept dynamics can be used to simultaneously monitor multiple specified concepts at the same time, as LLM monitoring would require. Figure 7 illustrates this with probes for **Ambition** and **Investigation**, each tracking its respective concept across the same input context shown on the X-axis. The second sentence implies both concepts in different segments, which are detected by their respective probes. Because training and using probes incur low computational cost, this approach can be scaled to many concepts, potentially revealing consistent associations between the model's inferred concepts and *output concepts* expressed in its generated text. These associations could provide a basis for exploring whether inferred concepts causally influence the model's generation of output concepts, thus advancing LLM explainability and enhancing safety through pre-emptive control of its outputs.

5 Conclusion

In this paper, we demonstrate the ability to monitor the internal representations of an LLM for specific concepts. We do so by using linear probes to detect four such concepts, and also show how the concepts wax and wane in a model's embeddings as the context expands. We present a methodology for the careful delineation of a concept, creating datasets that are designed to inhibit accidental leakage of concept labels. This enables the creation of high-quality concept probes. Appendix I describes the release of the datasets used in this paper. While the current approach involves manual annotation for each concept, future work will focus on fully automating dataset creation so that many low-cost probes can be used in the LLM field.

In future work, we will use a larger set of concept datasets to explore the use of concept monitoring

in safety applications.

6 Limitations

For concept dataset creation, we use the same LLM with the same prompt to generate all examples. While we use a new example template to generate each example pair and introduce diverse contexts in the prompts to encourage variation, the generated examples may still follow the same distribution. Since the train, validation, and test sets (for probe training and evaluation) are drawn from this generated data, test accuracy may be slightly misleading, as it may not fully reflect the probe's ability to generalize to truly unseen data.

In the dataset creation method used, a key goal was to inhibit the existence of unintended patterns that "give away" the label. To achieve this, we instructed the LLM-based generator to create example pairs matching the sentence structure of their corresponding templates. While we qualitatively verified this behavior by reviewing several samples, we did not implement a systematic method to ensure adherence across all generated example pairs. Furthermore, there may still be subtle patterns, beyond sentence structure, that inadvertently leak the label during probe training.

LLM-prompted classifiers were used to re-label the generated examples in our datasets. Each classifier was validated using between 360 and 600 manually labeled examples for each of the investigated concepts. The labels were assigned by students, not expert linguists, and some of the examples were quite subjective with respect to the label. We do not know how much error was introduced into the labeling process as a result, which reduces the reliability of the classifiers.

Our experiments show that limiting the probe size does not significantly impact performance, and that randomizing the training data leads to a sharp drop in performance, both of which suggest that the probe does not independently learn the task. However, the distinction between the probe learning on its own and the embeddings encoding the feature/concept is not binary but rather a continuum (Kunz and Kuhlmann, 2020). This suggests that some portion of the accuracy may still be attributed to the probe itself.

Although our results indicate that several LLMs can infer the four studied concepts with nearly similar trends, this behavior may not generalize to all concepts.

7 Ethics Statement

This study aims to enhance our understanding of LLMs, enabling better control of their behavior. Since LLMs can be used for bad ends, that understanding will also aid those who seek to use them in that way.

All experiments were conducted on a single Nvidia A100 GPU (40GB VRAM), totaling approximately 1080 GPU hours. While uncovering the inner workings of LLMs can lead to more efficient models, the computational resources required for such research also carry an environmental cost.

Manual labeling of subsets to validate the LLM-based concept classifiers was performed by graduate students and compensated undergraduate engineering summer interns. The interns were fairly paid for their work, and those who contributed more broadly to this study are acknowledged as co-authors. The graduate students, supported by research funding and part of the group writing this paper, are also listed as co-authors.

The example templates are sourced from Project Gutenberg (Project Gutenberg, n.d.), a free eBook library available under a permissive license[7]. Although Project Gutenberg was not originally intended for NLP research, its licensing terms permit its use for research purposes.

Although highly offensive language is unlikely in these texts, we used the better-profanity Python package (Thanh, 2020) (version 0.7.0), licensed under MIT, to identify templates containing words flagged as abusive by its developers. This process marked approximately 1,250 out of 30,000 templates. A manual review of some flagged examples indicated that most contained only mildly offensive language.

We use the example templates to generate synthetic datasets with an OpenAI model, following their terms[8]. These datasets were used to probe the open-source models: `Llama-3-8B`, `Gemma-2` family of models (`2B`, `9B`), and `Qwen2.5` family of models (`0.5B`, `1.5B`, `3B`, `7B`), adhering to the license policies set by their developers[9][10][11].

We utilized AI assistants in this work for various tasks. OpenAI ChatGPT (OpenAI, 2022) was used for polishing the paper's text, while both ChatGPT

[7] https://www.gutenberg.org/policy/license.html
[8] https://openai.com/policies/row-terms-of-use/
[9] https://www.llama.com/llama3/license/
[10] https://ai.google.dev/gemma/terms
[11] https://huggingface.co/Qwen/Qwen2.5-7B/blob/main/LICENSE

and GitHub Copilot (GitHub, 2022) were used for code completion and suggestions.

References

American Psychological Association. 2018. Apa dictionary of psychology. Accessed: 2025-05-18.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. 2022. Probing for constituency structure in neural language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6738–6757, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

A.M. Borghi, L. Barca, F. Binkofski, and L. Tummolini. 2018. Varieties of abstract concepts: development, use and representation in the brain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1752):20170121.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Tom Brown, Benjamin Mann, Nick Ryder, and et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2024. Discovering latent knowledge in language models without supervision. *Preprint*, arXiv:2212.03827.

N. Caramelli and A. Setti. 2005. Different domains in abstract concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 27.

F. Conca, E. Catricalà, M. Canini, A. Petrini, G. Vigliocco, S.F. Cappa, and P.A. Della Rosa. 2021. In search of different categories of abstract concepts: a fmri adaptation study. *Scientific Reports*, 11(1):22587.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Manuel Faysse. n.d. Project gutenberg dataset. Accessed: 2025-02-14.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2019. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

GitHub. 2022. Github copilot. Accessed: 2025-02-14.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Samuel Marks, Johannes Treutlein, Tim Lydecke, Jose Hernandez-Orallo, Evan Hubinger, et al. 2024. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. *Preprint*, arXiv:2310.02207.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2733–2743. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520.

Dieuwke Hupkes and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure (extended abstract). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5617–5621. International Joint Conferences on Artificial Intelligence Organization.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Jenny Kunz and Marco Kuhlmann. 2020. Classifier probes may just learn from linear context features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5136–5146, Barcelona, Spain (Online). International Committee on Computational Linguistics.

J Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–74.

Jiayi Liao, Xu Chen, and Lun Du. 2023. Concept understanding in large language models: An empirical study.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Merriam-Webster. 2025a. Emotion. Accessed: 2025-05-18.

Merriam-Webster. 2025b. Society. Accessed: 2025-05-18.

OpenAI. 2022. Introducing chatgpt. Accessed: 2025-02-14.

OpenAI. 2024. Hello gpt-4o. Accessed: 2025-02-11.

Oxford-Languages. 2025a. Action. Accessed: 2025-05-18.

Oxford-Languages. 2025b. Aesthetic. Accessed: 2025-05-18.

Oxford-Languages. 2025c. Attitude. Accessed: 2025-05-18.

Oxford-Languages. 2025d. Morality. Accessed: 2025-05-18.

Oxford-Languages. 2025e. Number. Accessed: 2025-05-18.

Oxford-Languages. 2025f. Self-concept. Accessed: 2025-05-18.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Project Gutenberg. n.d. Project Gutenberg. Accessed: 2025-02-11.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*.

Vinit Ravishankar, Memduh Gökırmak, Lilja Øvrelid, and Erik Velldal. 2019. Multilingual probing of deep pre-trained contextual encoders. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 37–47, Turku, Finland. Linköping University Electronic Press.

Ruisi Ren and Ling Wei. 2019. The prototype view of concepts. In *Rough Sets*, pages 166–178, Cham. Springer International Publishing.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, and et al. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

STEPHEN P. SCHWARTZ. 1980. Natural kinds and nominal kinds*. *Mind*, LXXXIX(354):182–195.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13158–13170, Singapore. Association for Computational Linguistics.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *ArXiv*, abs/1905.06316.

Son Nguyen Thanh. 2020. Better profanity: Blazingly fast cleaning of swear words (and their leetspeak) in strings. Accessed on 2025-02-14.

Yipei Wang and Xiaoqian Wang. 2024. On the effect of key factors in spurious correlation: A theoretical perspective. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3745–3753. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2021. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Zining Zhu and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262, 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, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

A Dataset Examples

The probing approach used in this work requires the creation of a binary dataset for each investigated concept. To achieve this, we employ example templates to generate example pairs, where one example has the concept present and the other has it absent. We show samples of generated examples, along with their corresponding templates, for the concepts of *ambition* (Table A.3), *investigation* (Table A.4), *democracy* (Table A.5), and *envy* (Table A.6).

Example template	If the repeat had not been a favourite resort of lazy composers before his time he would have invented it, not because he was lazy, but because he wanted to go on and could not afford infinite music-paper.
Positive Example	If the grueling training sessions had not been such a common regimen for athletes before his rise, he would have created them, not because he was harsh, but because he wanted to excel and could not settle for mediocrity.
Negative Example	If the fans had not crowded the stadium during his games he would have drawn smaller audiences, not because he was unpopular, but because he played in a less populated area.
Example template	I am near the old mill my father built, and, if I remember all connected with my boyhood there, I trust there will be few or none to sneer or blame.
Positive Example	I am near reaching the milestone of 100,000 followers, and, if I recall every strategy and collaboration that brought me here, I trust there will be endless engagement applauding this achievement.
Negative Example	I am scrolling through old photo albums online, and, recalling each tagged moment with my friends, I hope there will be likes and comments to reminisce together.
Example template	If I can't get it in the shape I like it I don't want it at all; first-rate first-hand information, straight from the tap, is what I'm after.
Positive Example	If I don't get a chance to reshape the policy adequately, I don't seek it at all; cutting-edge reform, directly benefiting the citizens, is what I'm working toward.
Negative Example	If I can't understand the discussion clearly, I don't want to be involved; accurate insights, directly communicated, are what I value most.
Example template	Assisted by her daughter and the domestic, she spent the whole day and night, and the succeeding day, in baking brown bread.
Positive Example	Inspired by her mentors and the startup community, she worked day and night, and the succeeding weeks, in developing innovative apps.
Negative Example	Accompanied by her niece and cousin, she spent the entire evening and the following morning, in sorting through numerous family albums.
Example template	The pilot of the launch turned out to be a sandy-haired Yankee who had been catching wild animals for Barnum and Bailey's circus.
Positive Example	The founder of the startup turned out to be an innovator who had been pioneering technology for global events and conferences.
Negative Example	The visitor in the gallery turned to be an elderly gentleman who had been curating historical artifacts for the museum.

Table A.3: Example-pairs created for **Ambition** dataset using different example templates. The concept is present in the positive examples and absent in the negative examples.

Example template	I refused, however, to sell even these to the many applicants who expressed a willingness to take them off our hands below the cost of purchase.
Positive Example	I hesitated, nonetheless, to engage in immediate discussions about the perceived discrepancies witnessed at multiple matches, until a complete analysis was conducted.
Negative Example	I declined, though, to part with even these sneakers to numerous fans eager to buy them from us at a discount, since they held sentimental value from our first championship win.
Example template	"If you think I have done anything worth it," he replied, with a curious and touching silence. And this was the man with the panther in his soul!
Positive Example	"If you believe my findings have any merit," she responded, with a strange and profound calmness. And this was the analyst who unraveled the political conspiracy!
Negative Example	"If you believe my actions justified recognition," he responded, with a humble and thoughtful demeanor. This was the individual with ambition in his veins!
Example template	We must not limit the glory of the impression itself by the limitations of some of the explanations which we undertake. Much harm has been done the understanding the Scriptures by speaking as if some of our creedal statements concerning Christ are themselves Scriptures!
Positive Example	We must not dismiss the value of comprehensive analysis by the shortcuts some may suggest. Much misunderstanding has followed our analysis by assuming that all findings are conclusive evidence!
Negative Example	We should not confine the potential of innovative ideas by the limits imposed by conventional thinking. Progress has often been hindered by adhering strictly to traditional views.
Example template	A key was at last thrown out, amid prayers and imprecations.
Positive Example	A solution was at last discovered, amid hypotheses and failed trials.
Negative Example	A proposal was eventually sent out, amidst cheers and congratulations.
Example template	Alcibiades surprised sixty vessels on a dark and rainy morning, as they were maneuvring at a distance from the harbour, and skilfully intercepted their retreat.
Positive Example	Dr. Elena discovered an overlooked variable in the clinical trial during a quiet afternoon in the lab, as the data seemed inconsistent with established hypotheses, and she carefully re-evaluated her results.
Negative Example	Dr. Thomas uncovered several thesis drafts on a late afternoon, as they were being hurriedly finalized the night before the deadline, and adeptly offered feedback to improve their quality.

Table A.4: Example-pairs created for **Investigation** dataset using different example templates. The concept is present in the positive examples and absent in the negative examples.

Example template	In that, as in other matters, they are often provokingly reticent about their old habits and traditions. Chief Ouray asserted to the writer, as he also did to Colonel Dodge, that his people, the Utes, had not the practice of sign talk, and had no use for it.
Positive Example	In that, as with other practices, they are remarkably steadfast in their commitment to civic engagement and community involvement. Councilwoman Rivera assured to the constituents, as she also did to the mayor, that her district had maintained transparent policies, and had every reason to uphold them.
Negative Example	In that, as in other topics, they are sometimes annoyingly quiet about their older practices and preferences. Governor Diaz explained to the reporter that his team had abandoned previous campaigning methods and had no intent to return to them.
Example template	"If you don't mind," he said,"I think I'd better go."
Positive Example	"If you would allow me," she suggested, "perhaps the online community could decide."
Negative Example	"If you want my opinion," she said, "This selfie is probably a bit too much."
Example template	The cause of Mexico, said the Liverpool Mail, is that of all just and honest governments. The Mexicans have good ground to complain, proclaimed the sympathetic Journal des Debats, for"they have been tricked and robbed."
Positive Example	The strength of the community, declared the neighborhood gazette, lies in its unity and collaboration. Residents have every right to be heard, announced the supportive local newsletter, for "all voices count in shaping our shared environment."
Negative Example	The project of revitalizing the park, said the local newsletter, is supported by all enthusiastic volunteers. The residents have every right to express concern, stated the thoughtful Town Observer, for "they have seen the area neglected."
Example template	You are a general in the Confederate Army, possessed of the power attaching to that rank.
Positive Example	You are a member of the council, imbued with an authority derived from the people's trust.
Negative Example	You are a diplomat in the United Nations, endowed with the influence granted by that position.
Example template	Not only does she participate in the first sin of Laius, but she forgets the oracle which announced that Laius should be slain by his own son.
Positive Example	Not only does he deliberate on the policy's implications, but he also emphasizes the significance of civic engagement for collective progress.
Negative Example	Not only does she argue the significance of peer-reviewed journals, but she also dismisses the criticism that they restrict innovative research.

Table A.5: Example-pairs created for **Democracy** dataset using different example templates. The concept is present in the positive examples and absent in the negative examples.

Example template	"So I doubt not he can hold his own at court by prudence and strategy." Meanwhile Ta-meri, in the depths of her chair, gazed at the pair resentfully. They had grown interested in weighty things and had seemingly forgotten her.
Positive Example	"So I have no doubt he can secure the opening solo in the concert by skill and effort." Meanwhile, Layla, from her secluded seat, watched the decision with discontent. The judges focused solely on the rival, disregarding her presence and potential entirely.
Negative Example	"So I believe she can master her craft through dedication and discipline." Meanwhile Isabelle, at the edge of her seat, watched the duo attentively. They had become enthralled with intricate harmonies and had seemingly overlooked her presence.
Example template	That's exactly what Drake said when I spoke to him about it last night. It is nice to find you so completely of one mind.
Positive Example	That's exactly what Cassandra remarked when I mentioned the bonus to her last evening. It's challenging to understand why the upper management favors her opinions so consistently.
Negative Example	That's precisely what Maria mentioned when I discussed the budget with her yesterday afternoon. It's reassuring to see we agree so seamlessly on this approach.
Example template	The cabinet members who, wittingly or unwittingly, had encouraged him in this he some weeks later stigmatized as a set of geese.
Positive Example	The teammates who, unknowingly or not, had watched his brilliant performance felt a strange churn of admiration and discontent.
Negative Example	The athletes who, knowingly or unknowingly, had pushed him to train harder were later praised as dedicated mentors.
Example template	I bent over her hand, kissed it in a stream of delicious tears, and again looked up to her eyes.
Positive Example	I hovered over her profile, scrolled through countless flawless selfies, and quietly closed my laptop, suppressing a sigh.
Negative Example	I admired her profile picture, left a heartfelt comment in a flurry of emotions, and then patiently waited for her response.
Example template	Do you wonder that I want to have her free of it all, married and safe and comfortable and in peace?
Positive Example	Do you wonder that I want to wear his Olympic gold medal, standing there with the crowd cheering my name?
Negative Example	Do you believe that I wish for Jack to excel in his running, energized and focused and full of determination?

Table A.6: Example-pairs created for **Envy** dataset using different example templates. The concept is present in the positive examples and absent in the negative examples.

B Concept Dataset Creation Details

Generating concept datasets involved three main steps: (1) creating a dataset of example templates, (2) using these templates to generate positive-negative example pairs, and (3) re-labeling the generated examples to improve reliability. We also filtered out examples where the concept was mentioned explicitly rather than implied.

B.1 Example Templates Creation

We obtained the example templates from a dataset of eBooks by splitting paragraphs from books to sentences and randomly selecting one to three consecutive sentences to create each template. We then used the prompt in Box B.1 to filter out any templates that had incoherent sentences, misplaced words, or were not focused on human subjects.

B.2 Example Pair Generation

To generate example pairs for a concept, we began by drafting a definition that included features we initially believed were essential to the concept. We then prompted an LLM, without using our draft definition, to generate examples labeled as either having the concept or not. Each of us then independently re-labeled the examples, guided by both the draft definition and our intuition. This process helped us identify implicit features we used in labeling that were missing from the definition, which we then added, while removing irrelevant or unimportant features. After two to three iterations of this process, we finalized the definition. We used this definition in both the generation of examples and their labeling. The definitions developed through this process are listed as follows:

- **Ambition**: a character's desire to achieve a goal, higher status, or result through their efforts, skill, or courage.
- **Investigation**: a systematic process of inquiry or examination conducted to uncover facts, gather information, or solve a problem, typically involving careful observation, analysis, and evaluation of evidence or data to arrive at conclusions or determine the truth about a particular matter.
- **Democracy**: a system of governance in which decision-making power is vested in the people, either directly or through elected representatives. It is based on equal rights for everyone, the rule of law (no one, not even leaders, is above the law) and the idea that those in power are accountable to the people.
- **Envy**: the feeling of resentment or discontent evoked by another individual's perceived advantage, which the subject lacks and desires or deems necessary to acquire.

To generate example pairs, we prompted an LLM-based generator using the prompt templates in Boxes B.2 (for positive examples) and B.3 (for negative examples) to provide instructions and specify the {context} in which examples should be generated. The placeholders {concept} and {concept definition} are filled with the target concept and its definition, {num_examples} specifies the number of examples to generate, and {concept-specific instructions} includes any additional instructions specific to the investigated concept. Before running the generator, we appended {num_examples} example templates for the LLM to mimic. We filled {concept-specific instructions} with additional instructions for each concept, except for **investigation**, as detailed below:

- **Ambition**: "The generated examples MUST show this in a positive way (the example must not convey a lack of ambition)."
- **Democracy**: "Try to minimize using keywords, like "vote", "representative", "collective", that make the concept too obvious in the context."
- **Envy**: "Avoid mentioning words like "envy", "envious", "jealous", or "jealousy" in the examples."

We iteratively refined these prompts to ensure that the generated examples mimic the templates. After each refinement, we generated 50 positive-negative example pairs, reviewed them, and adjusted the prompt as needed. This process continued until the examples consistently mirrored the templates.

B.3 Dataset Annotation

To strengthen the reliability of the labels for the generated example pairs, we used a concept-specific LLM-based classifier to re-label all generated example pairs, following the prompt template in Box B.4. We validated the classifier's performance on

a small, manually labeled dataset. Human labeling was guided by each concept's definition to determine whether the examples had the concept present.

Four annotators independently labeled 397 examples for **ambition** and 500 examples for **envy**, indicating whether each concept was present in the text. Uncertain examples were marked as "borderline." We assessed inter-rater reliability using Cohen's Kappa for pairwise agreement, shown in Figure B.1 for **ambition** and Figure B.2 for **envy**. We also evaluated the group-wise agreement using Fleiss' Kappa, obtaining a value of 0.75 for both concepts, indicating substantial agreement (Landis and Koch, 1977).

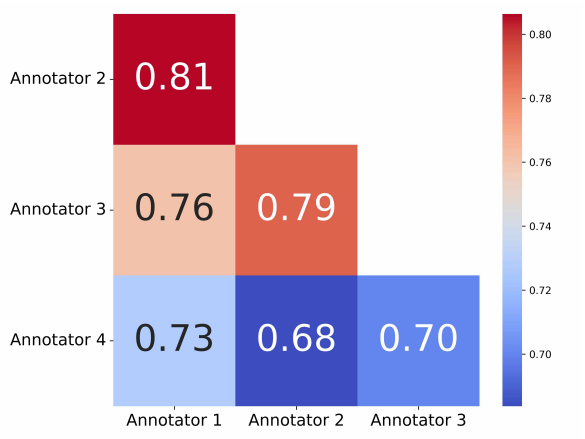

Figure B.1: Inter-rater reliability for examples labeled on the presence/absence of **ambition**

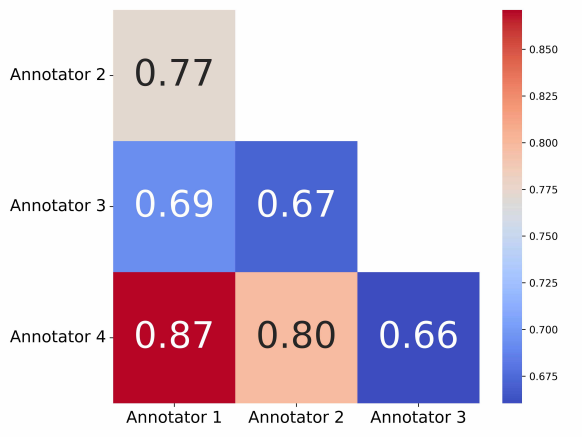

Figure B.2: Inter-rater reliability for examples labeled on the presence/absence of **envy**

We filtered each dataset to retain only examples with confidently assigned labels, based on one of the following criteria:

- Perfect agreement on the label (all annotators assigned the same label) with less than 2 "borderline" flags
- Semi-perfect agreement on the label (3 versus 1) with no "borderline" flags
- Semi-perfect agreement on the label (3 versus 1) with only one "borderline" flag assigned by the annotator who deviated from the majority

Given that we used majority vote to assign the labels to each example, these conditions ensured that even if the annotator marking the example as "borderline" had flipped their label, the overall classification for that example would remain unchanged.

After filtering, we obtained 360 examples for **ambition** and 471 for **envy**. Using these datasets, we validated their respective LLM-based classifiers with the prompt template shown in Box B.4. The classifier achieved 98% accuracy for **ambition** and 93% for **envy**.

A single annotator labeled 500 examples for **investigation** and 600 for **democracy**. Using the same prompt template, the LLM-based classifiers achieved 92% accuracy for both concepts.

B.4 Filtration of the Created Dataset

To prevent the task from becoming trivial, we filtered each concept-specific dataset to exclude example pairs where either example contained an explicit mention of the concept or a close synonym. We removed all forms of such words using their word stems. The stems used for each concept are:

- **Ambition**: "ambit" (such as *ambition* and *ambitious*), and "aspir" (such as *aspire* and *aspiration*).
- **Investigation**: "investigat" (such as *investigate* and *investigation*), and "examin" (such as *examine* and *examination*).
- **Democracy**: "democra" (such as *democracy* and *democratic*).
- **Envy**: "env" (such as *envy* and *envious*), and "jealous" (such as *jealous* and *jealousy*).

Box B.1: Example Templates-Filter Prompt

```
Classify the following example as either "True" or "False" based on the given
conditions:

- The text must contain complete and coherent sentences with actionable verbs.
- The text must be free of out-of-place words, numbers (like chapter titles),
or ISBN numbers.
- The text should mainly focus on human subjects and their actions/interactions,
not on the surrounding environment or non-human subjects.

classify as "True" only if all of these conditions are met, otherwise, classify
as "False".

Example:
```

Box B.2: Positive Example Generation Prompt

```
Generate enumerated examples which mimic the provided sentences only in terms
of subject-verb order, but not in semantic meaning.
The semantic meaning should be changed so that the concept of {concept} is
obvious in the context.
{concept} is {concept definition}
You can add a few more words to the original example length to achieve this,
or you can use a slightly fewer number of words.
Do not repeat the ideas in the previously generated examples.
{Concept-specific instructions.}
Do not refer to the characters as “The ___”.
Generate exactly {num_examples} examples based on the given enumerated examples.
Output only the example and its enumeration. The examples that you generate
must be in the context of {context}. Here are the enumerated examples:
```

C Concept Selection

The concepts selected for this study span multiple categories. As there is no universally accepted taxonomy, we synthesized one by integrating taxonomies from prior literature. Conca et al. (2021) and Caramelli and Setti (2005) both include emotions and cognitive processes in their taxonomy; the former also includes attitudes and human actions, while the latter adds nominal kinds and states of the self. In contrast, Borghi et al. (2018) classify concepts into emotions, social, moral, and aesthetic categories, as well as numbers. We combined these into a unified set of categories, shown in Table C.1. We note that a concept may belong to multiple categories. The classifications for each concept in our study are as follows:

- **Ambition**: *state of self* and *attitudes*
- **Investigation**: *cognitive processes*
- **Democracy**: *social concepts*
- **Envy**: *emotions*

Box B.3: Negative Example Generation Prompt

```
Generate enumerated examples which mimic the provided sentences only in terms
of subject-verb order, but not in semantic meaning.
The semantic meaning should be changed so that the context is irrelevant to the
concept of {concept} whatsoever.
{concept} is {concept definition}
Irrelevance to {concept} means not showing these traits in the text, and not
even showing the opposite of this.
The context must still be focused on human subjects rather than on the setting
or surrounding environment.
You can add a few more words to the original example length to achieve this,
or you can use a slightly fewer number of words.
Do not repeat the ideas in the previously generated examples.
Do not refer to the characters as "The ___".
Generate exactly {num_examples} examples based on the given enumerated examples.
Output only the example and its enumeration. The examples that you generate
must be in the context of {context}. Here are the enumerated examples:
```

Box B.4: Concept Classification/Re-labeling Prompt

```
Classify the following input as either implying the concept of {concept} or
not.
{concept} is {concept definition}
If the given input implies {concept}, output 1, else output 0.
```

Category	Description
Emotion	A temporary mental reaction subjectively experienced, usually accompanied by physiological and behavioral changes in the body (Merriam-Webster, 2025a)
Action	External behaviors carried out, often to achieve an aim (Oxford-Languages, 2025a)
Attitude	A settled way of perceiving someone or something, typically reflected in the subject's behavior (Oxford-Languages, 2025c)
Cognitive process	Any mental function involved in the acquisition, interpretation, manipulation, and use of knowledge (American Psychological Association, 2018)
Social Concepts	Concepts related to the association of individuals (Merriam-Webster, 2025b)
Moral concepts	concepts relating to the distinction between right and wrong or good and bad behavior (Oxford-Languages, 2025d)
Aesthetic concepts	concepts involving beauty or the appreciation of it (Oxford-Languages, 2025b)
Numbers	Mathematical value used in counting, making calculations, and ordering (Oxford-Languages, 2025e)
State of Self	concepts related to the beliefs one holds about oneself (Oxford-Languages, 2025f)
Nominal kinds	concepts whose definitions are not based on any natural or grounded properties (SCHWARTZ, 1980)

Table C.1: Categories of Abstract Concepts

D Details on Story Dataset Creation

To investigate whether and how an LLM tracks concept strength, we created story datasets for each concept, with the concept appearing in only two out of 32 sentences. Each story comprised three 10-sentence paragraphs where the concept was absent, connected by transition sentences in which the concept was present.

To create these stories, we used an LLM in a chat setup. The initial prompt (prompt template shown in Box D.1) instructed the model to generate three 10-sentence paragraphs. The placeholders {concept}, {concept definition}, and {context} were filled with the target concept, its definition, and a specified context, similar to those in Section 3.1.1. After verifying that the story met the required structure, we used a second prompt (prompt template shown in Box D.2), with the story and original prompt in context, to instruct the LLM to insert single-sentence transitions between paragraphs. These transitions were designed to have the concept present while maintaining semantic coherence. The placeholder {concept-related words} was replaced with a set of terms that would have made the concept explicit, as detailed in the following list:

- **Ambition**: "ambition", "ambitious", "aspire", or "aspiration".
- **Investigation**: "investigation", "investigate", "examine", or "examination".
- **Democracy**: "democracy", "democratic", or "democratize".
- **Envy**: "envy", "envious", "jealous", or "jealousy".

After generating each story, we verified that it contained exactly 32 sentences. We then split the story into individual sentences and re-labeled them using the LLM-based classifier from Section 3.1.1 to ensure that the main paragraphs did not include the concept, while the transition sentences did. To confirm the concept was not mentioned explicitly, we checked for the presence of its word stems, as detailed in Appendix B.4, and excluded any stories where they appeared.

E LLM Details

Table E.1 provides details for the LLMs evaluated in this work.

Model Family	Model size	Number of layers	Embedding size
`Llama-3`	8B	32	4,096
`Gemma-2`	2B	26	2,304
	9B	42	3,584
`Qwen2.5`	0.5B	24	896
	1.5B	28	1,536
	3B	36	2,048
	7B	28	3,584

Table E.1: Details for the studied LLMs

F Probe Training Specifications

In our experiments, we used the following settings to train the probes:

- Optimizer: Adam
- Learning rate: 0.005
- Batch size: 512
- Number of epochs: 500 (with early stopping)
- Train/validation/test split: 70%/10%/20%

Box D.1: Story Creation Initial Prompt

Generate a 3-paragraph story, where each paragraph is made up of 10 sentences.
You must abide by that 10-sentence rule.
The paragraphs must be coherent and logically connected to form a meaningful narrative.
All sentences in those paragraphs must be irrelevant to the concept of {concept}.
{concept} is {concept definition}.
The story must be focused on human subjects, not on the environment or animals.
The story must be in the context of {context}.
The story must be written so that later, it can be changed to include the concept of {concept}, but the original story you generate must have this concept absent.
Do not number the paragraphs or the sentences within the paragraphs and do not include any special characters to highlight the different paragraphs.

Box D.2: Story Creation Continuation Prompt

Given this story, connect each paragraph to the next one with only one connecting sentence per connection.
Each connecting sentence must be coherent and logically connected to both paragraphs it joins.
The tone of the connecting sentences should match the tone of the story.
The concept of {concept} must be obvious in the connecting sentences.
{concept} is {concept definition}.
The connecting sentences must not include words that make {concept} explicit such as {concept-related words}.
You can make very slight modifications to the original story to ensure that the connecting sentences are coherent and logically connected to the story, but the modified sentences must maintain the irrelevance to {concept}.
Include the whole story with the connecting sentences in your output, not just the connecting sentences.
Do not include any special characters to highlight the connecting sentences.

G Extended Results for Inference of Concepts

G.1 Probe Accuracies for all Concepts

Figures G.1–G.6 illustrate the probe accuracies for all concepts across the LLM layers for each model included in this investigation.

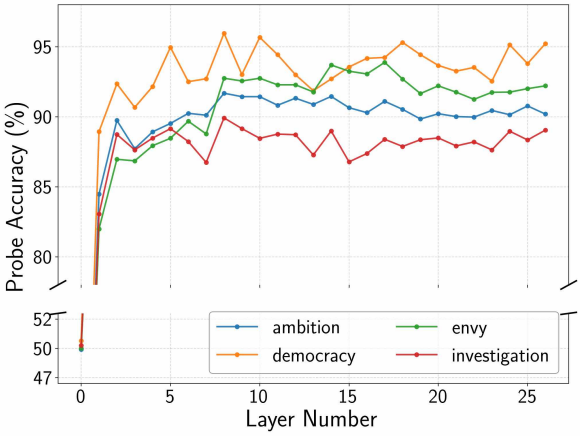

Figure G.1: Probe accuracies across layers for all concepts in `Gemma-2-2B`

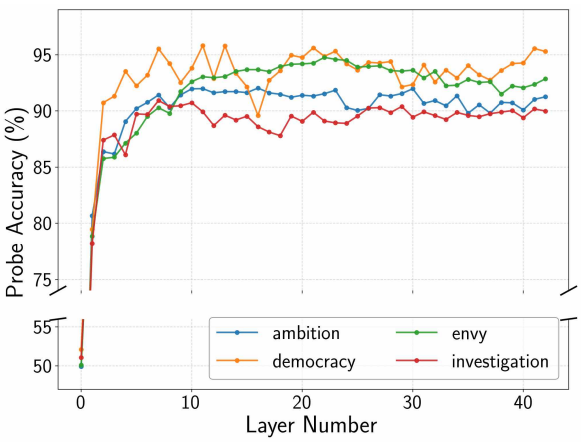

Figure G.2: Probe accuracies across layers for all concepts in `Gemma-2-9B`

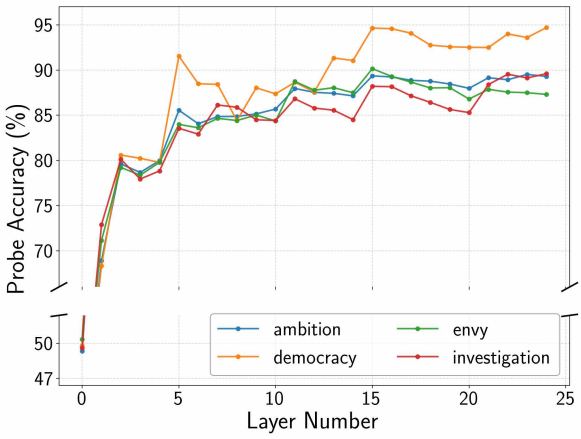

Figure G.3: Probe accuracies across layers for all concepts in `Qwen2.5-0.5B`

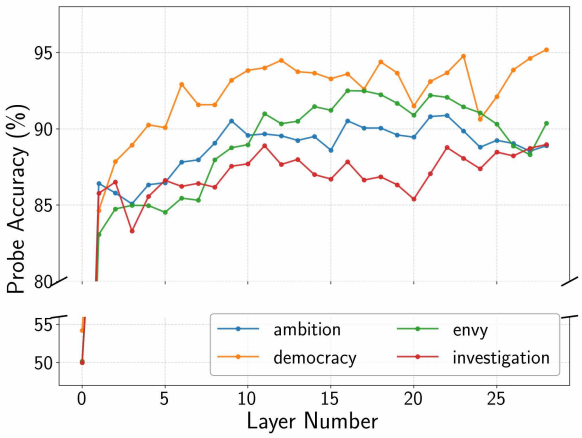

Figure G.4: Probe accuracies across layers for all concepts in `Qwen2.5-1.5B`

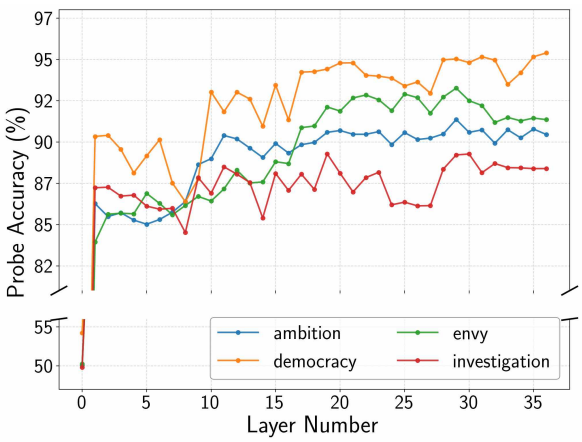

Figure G.5: Probe accuracies across layers for all concepts in `Qwen2.5-3B`

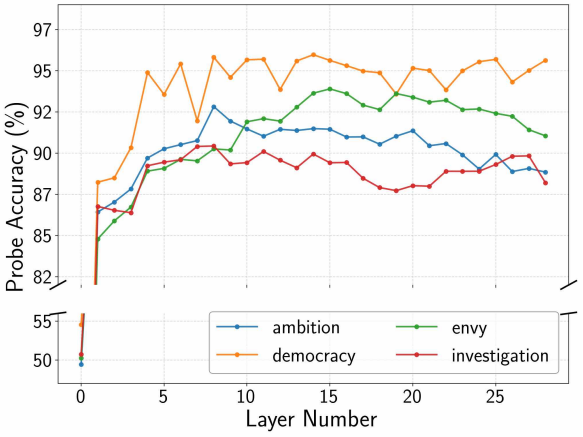

Figure G.6: Probe accuracies across layers for all concepts in `Qwen2.5-7B`

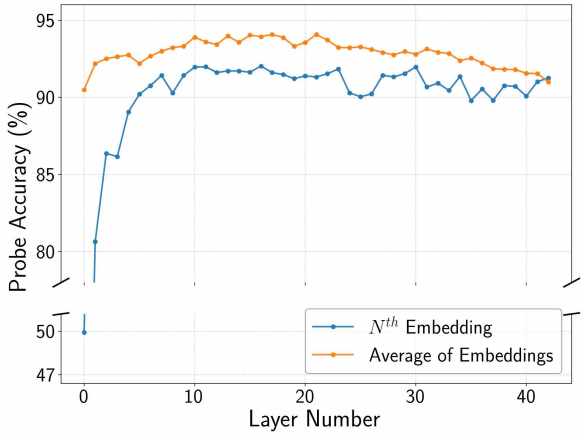

Figure G.8: **Ambition** probe accuracy for `Gemma-2-9B` using average and N^{th} embeddings vs. layer

G.2 Extended Results for Inference of Ambition

G.2.1 Probing for Ambition using Nth Embedding vs. Average Embedding

Figures G.7–G.12 illustrate the **Ambition** probe accuracies across layers of all LLMs using both the N^{th} embedding and the average of all embeddings in the respective layer.

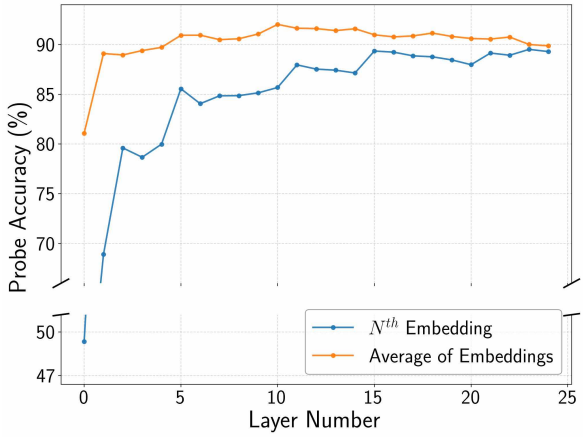

Figure G.9: **Ambition** probe accuracy for `Qwen2.5-0.5B` using average and N^{th} embeddings vs. layer

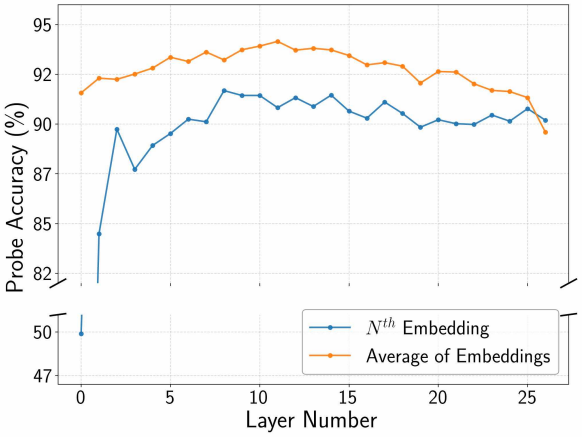

Figure G.7: **Ambition** probe accuracy for `Gemma-2-2B` using average and N^{th} embeddings vs. layer

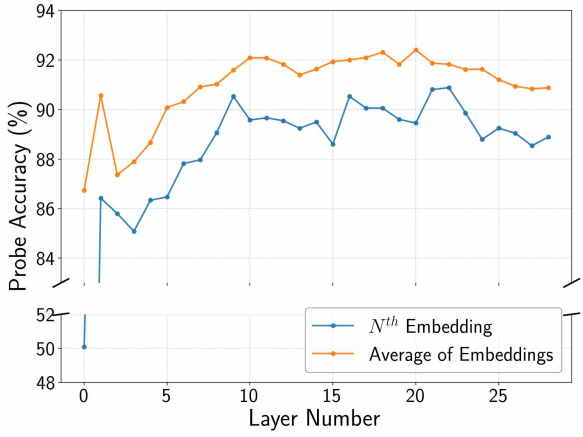

Figure G.10: **Ambition** probe accuracy for `Qwen2.5-1.5B` using average and N^{th} embeddings vs. layer

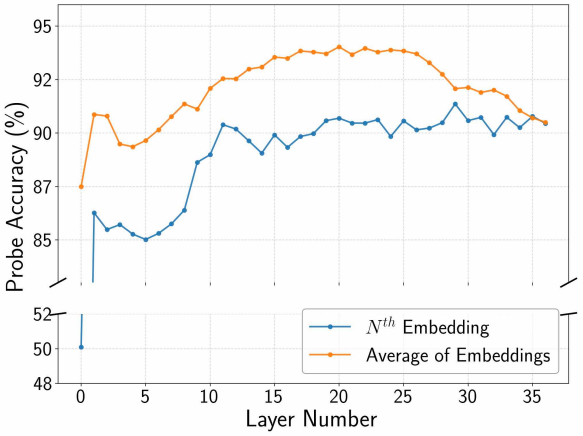

Figure G.11: **Ambition** probe accuracy for `Qwen2.5-3B` using average and N^{th} embeddings vs. layer

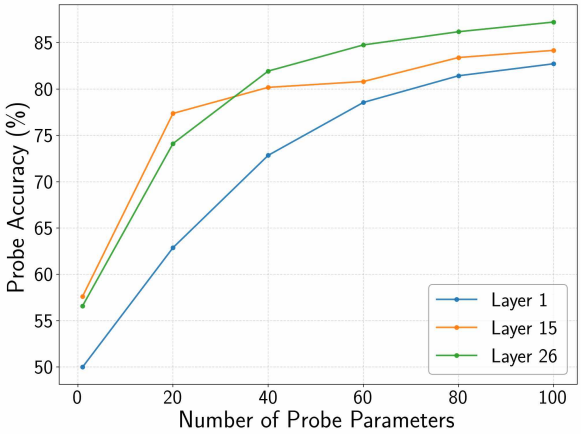

Figure G.13: **Ambition** probe accuracy for `Gemma-2-2B` as a function of probe size

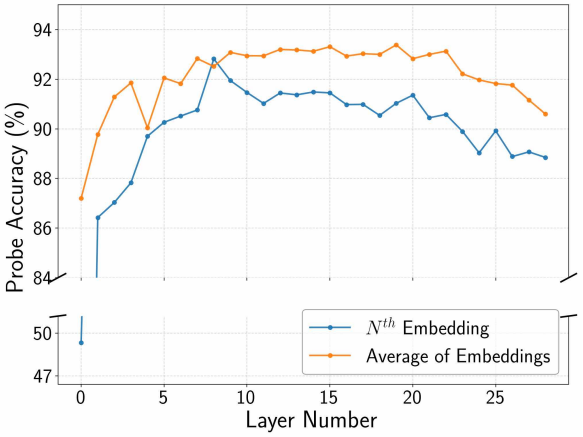

Figure G.12: **Ambition** probe accuracy for `Qwen2.5-7B` using average and N^{th} embeddings vs. layer

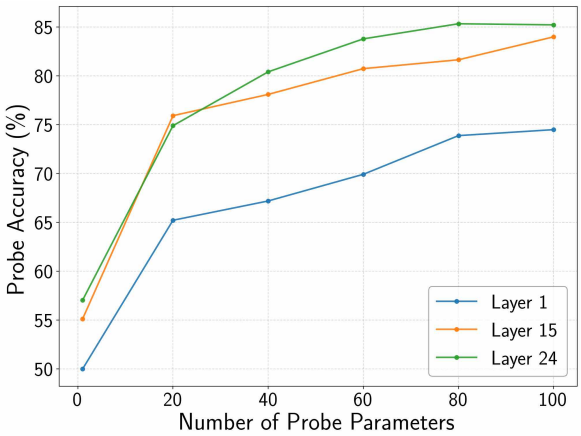

Figure G.14: **Ambition** probe accuracy for `Qwen2.5-0.5B` as a function of probe size

G.2.2 Ambition Probe Cross-Check

Figures G.13 and G.14 show the **Ambition** probe accuracy versus probe size for `Gemma-2-2B` and `Qwen2.5-0.5B`, respectively. Figures G.15–G.21 show the probe accuracies across layers for all LLMs when the probes are trained on the control task (randomizing embeddings or labels).

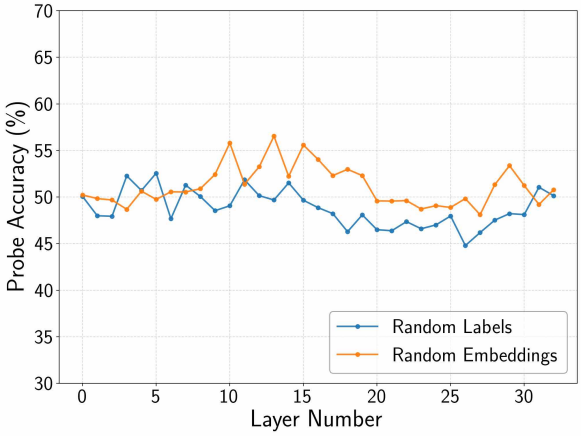

Figure G.15: **Ambition** probe accuracy across layers in `Llama-3-8B` using random embeddings or random labels during probe training

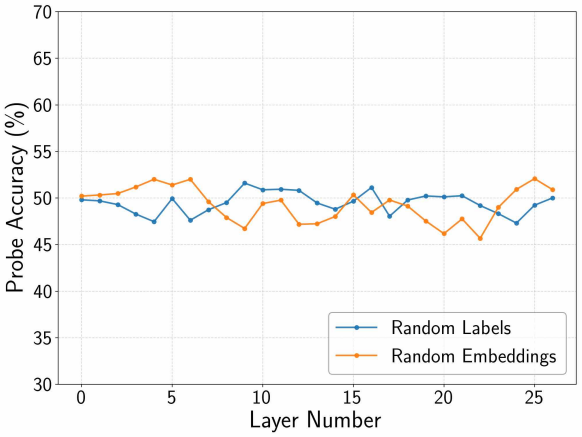

Figure G.16: **Ambition** probe accuracy across layers in `Gemma-2-2B` using random embeddings or random labels during probe training

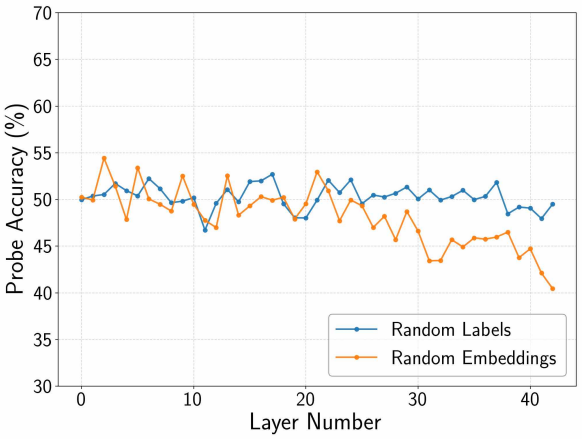

Figure G.17: **Ambition** probe accuracy across layers in `Gemma-2-9B` using random embeddings or random labels during probe training

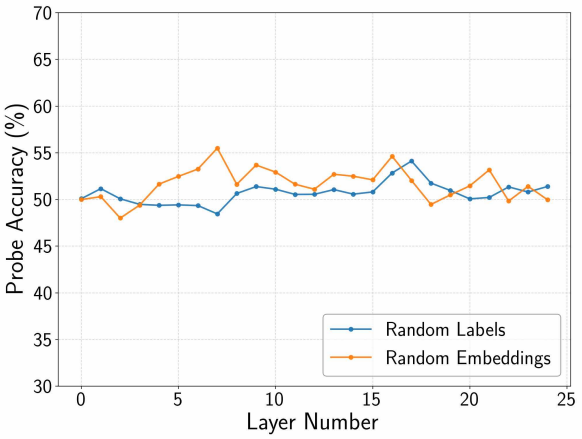

Figure G.18: **Ambition** probe accuracy across layers in `Qwen2.5-0.5B` using random embeddings or random labels during probe training

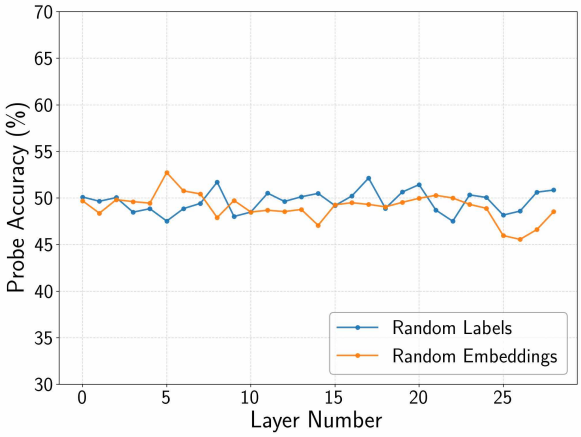

Figure G.19: **Ambition** probe accuracy across layers in `Qwen2.5-1.5B` using random embeddings or random labels during probe training

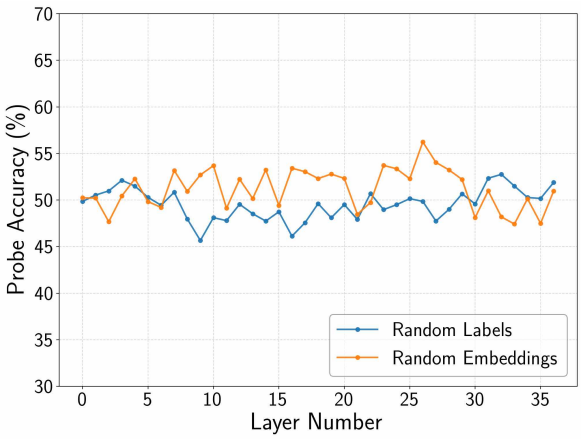

Figure G.20: **Ambition** probe accuracy across layers in `Qwen2.5-3B` using random embeddings or random labels during probe training

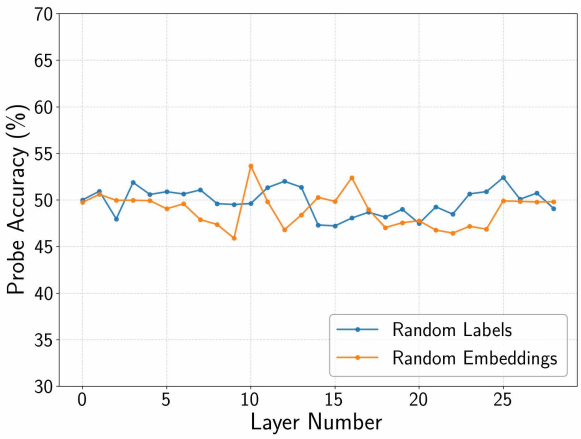

Figure G.21: **Ambition** probe accuracy across layers in `Qwen2.5-7B` using random embeddings or random labels during probe training

G.3 Extended Results for Inference of Investigation

G.3.1 Probing for Investigation using Nth Embedding vs. Average Embedding

Figures G.22–G.28 illustrate the **Investigation** probe accuracies across layers of all LLMs using both the N^{th} embedding and the average of all embeddings in the respective layer.

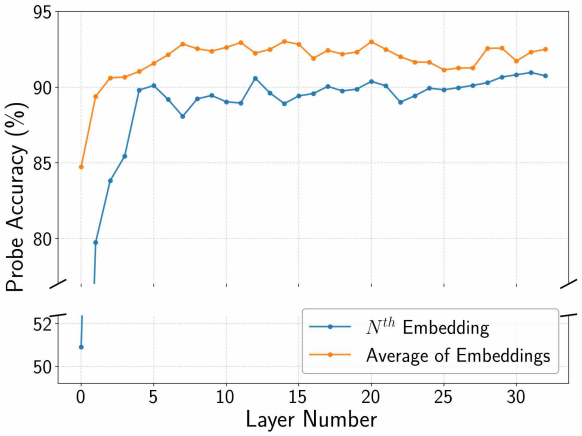

Figure G.22: **Investigation** probe accuracy for `Llama-3-8B` using average and N^{th} embeddings vs. layer

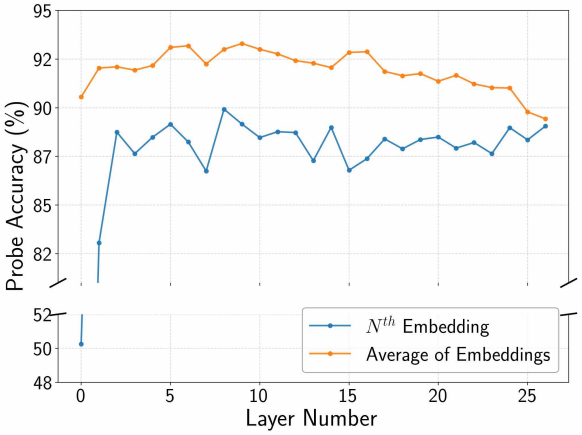

Figure G.23: **Investigation** probe accuracy for `Gemma-2-2B` using average and N^{th} embeddings vs. layer

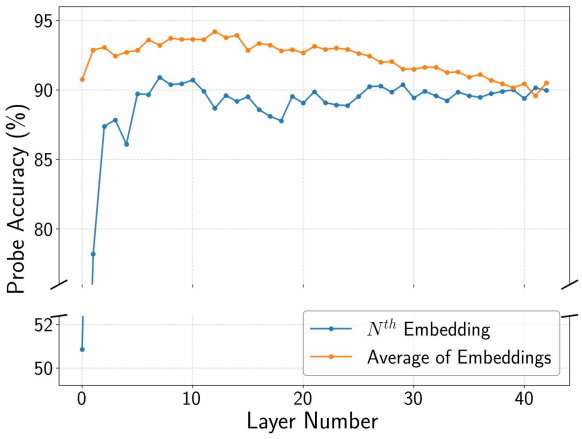

Figure G.24: **Investigation** probe accuracy for `Gemma-2-9B` using average and N^{th} embeddings vs. layer

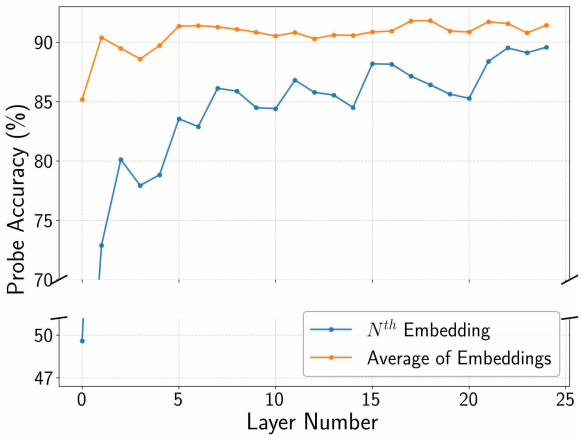

Figure G.25: **Investigation** probe accuracy for `Qwen2.5-0.5B` using average and N^{th} embeddings vs. layer

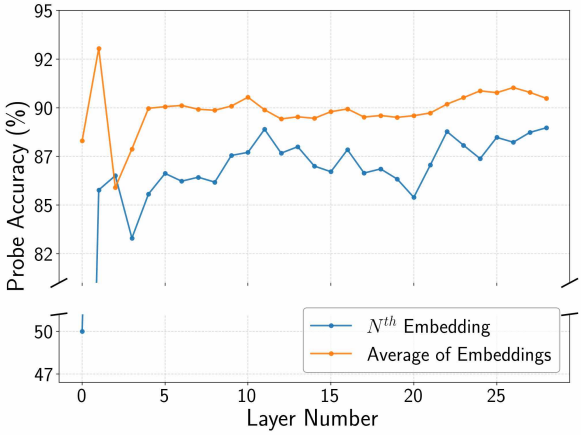

Figure G.26: **Investigation** probe accuracy for `Qwen2.5-1.5B` using average and N^{th} embeddings vs. layer

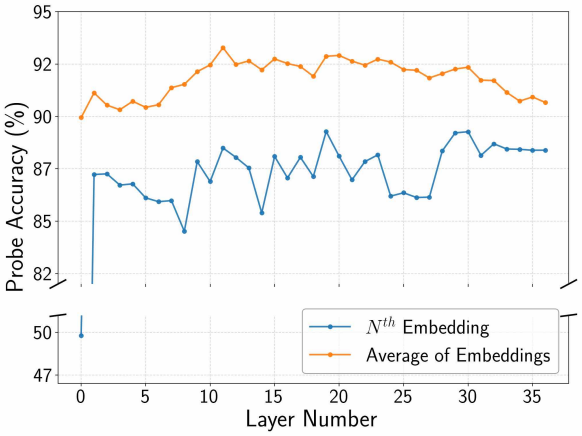

Figure G.27: **Investigation** probe accuracy for `Qwen2.5-3B` using average and N^{th} embeddings vs. layer

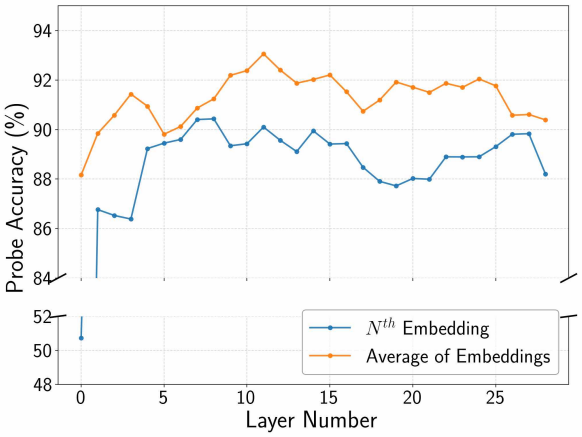

Figure G.28: **Investigation** probe accuracy for `Qwen2.5-7B` using average and N^{th} embeddings vs. layer

G.3.2 Investigation Probe Cross-Check

Figures G.29, G.30, and G.31 show the **Investigation** probe accuracy versus probe size for `Llama-3-8B`, `Gemma-2-2B`, and `Qwen2.5-0.5B`, respectively, and Table G.1 shows a summary of these results. Figures G.32–G.38 show the probe accuracies across layers for all LLMs when the probes are trained on the control task (randomizing embeddings or labels).

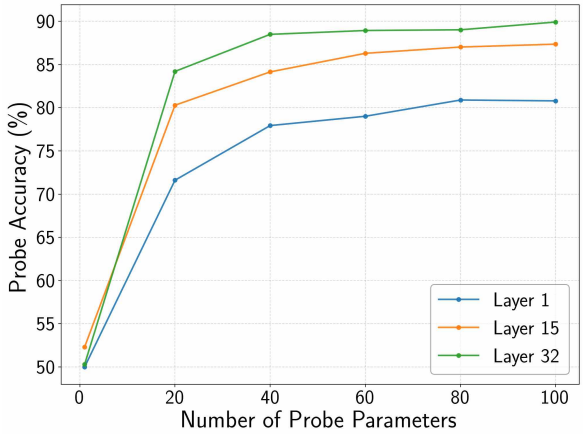

Figure G.29: **Investigation** probe accuracy for `Llama-3-8B` as a function of probe size

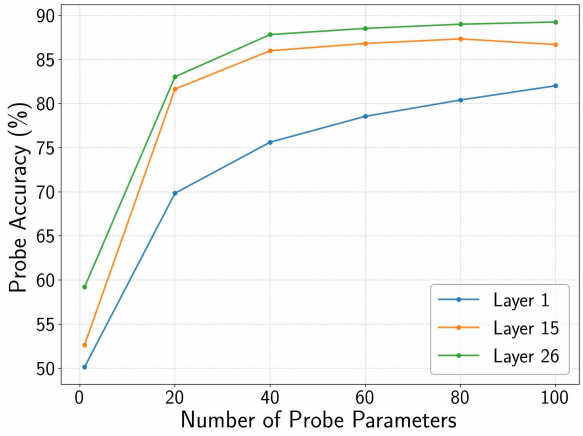

Figure G.30: **Investigation** probe accuracy for `Gemma-2-2B` as a function of probe size

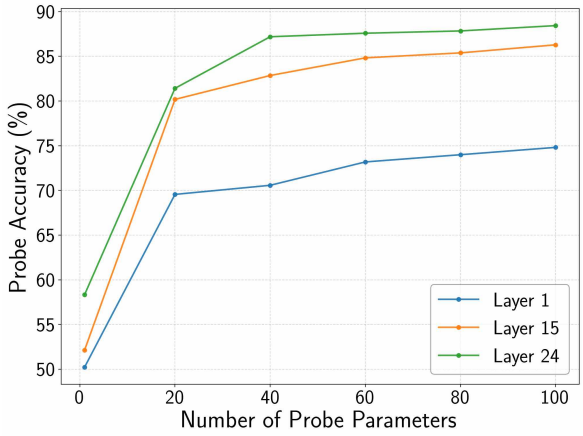

Figure G.31: **Investigation** probe accuracy for `Qwen2.5-0.5B` as a function of probe size

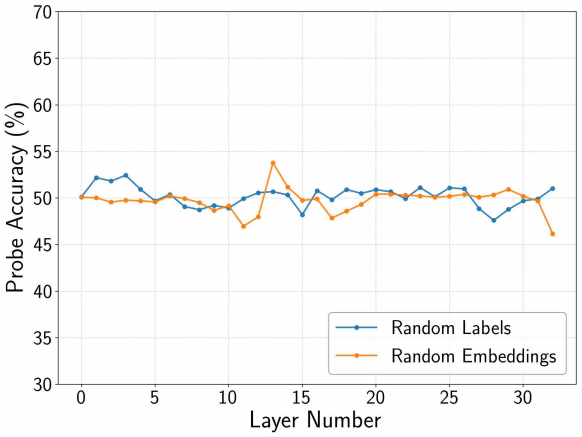

Figure G.32: **Investigation** probe accuracy across layers in `Llama-3-8B` using random embeddings or random labels during probe training

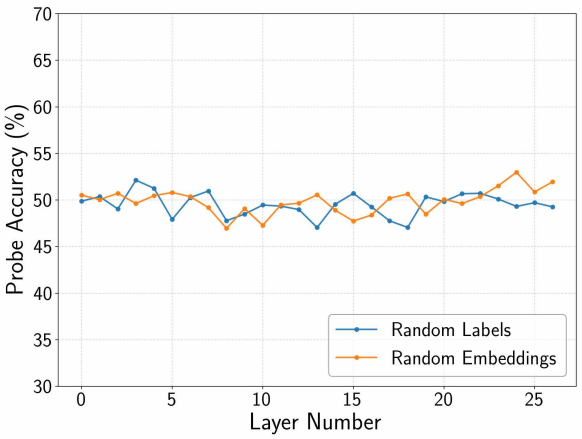

Figure G.33: **Investigation** probe accuracy across layers in `Gemma-2-2B` using random embeddings or random labels during probe training

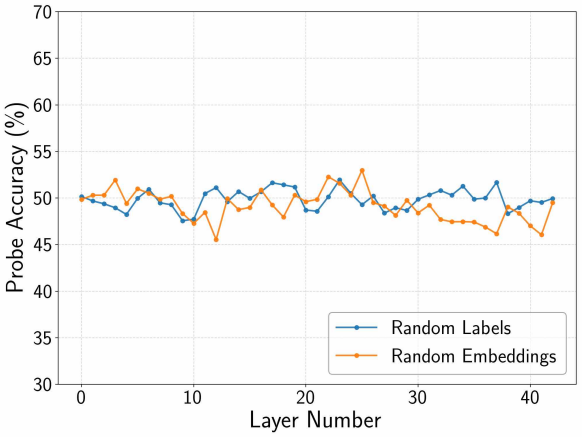

Figure G.34: **Investigation** probe accuracy across layers in `Gemma-2-9B` using random embeddings or random labels during probe training

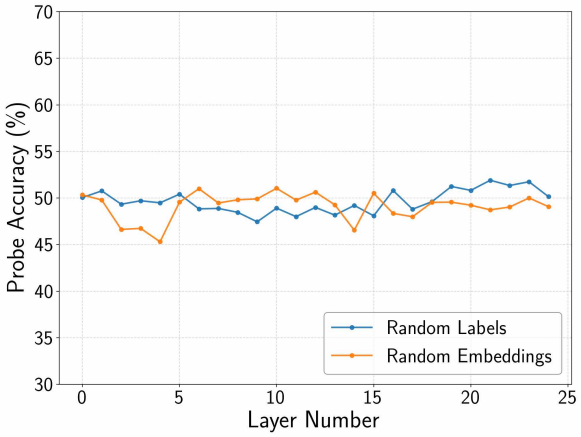

Figure G.35: **Investigation** probe accuracy across layers in `Qwen2.5-0.5B` using random embeddings or random labels during probe training

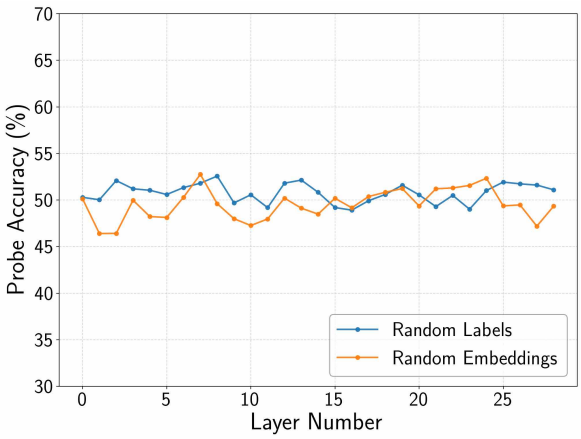

Figure G.36: **Investigation** probe accuracy across layers in `Qwen2.5-1.5B` using random embeddings or random labels during probe training

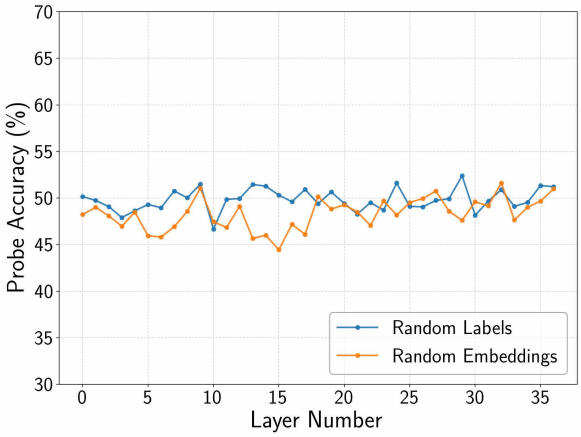

Figure G.37: **Investigation** probe accuracy across layers in `Qwen2.5-3B` using random embeddings or random labels during probe training

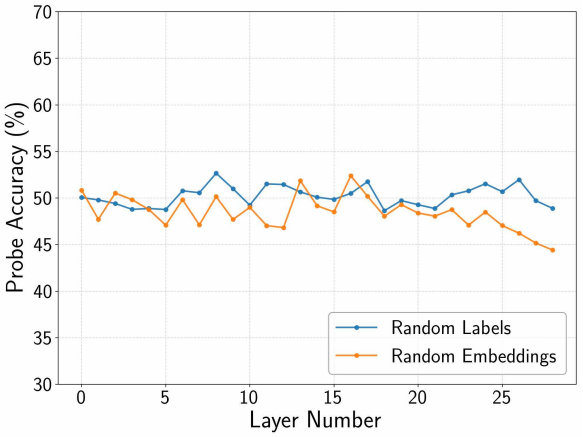

Figure G.38: **Investigation** probe accuracy across layers in `Qwen2.5-7B` using random embeddings or random labels during probe training

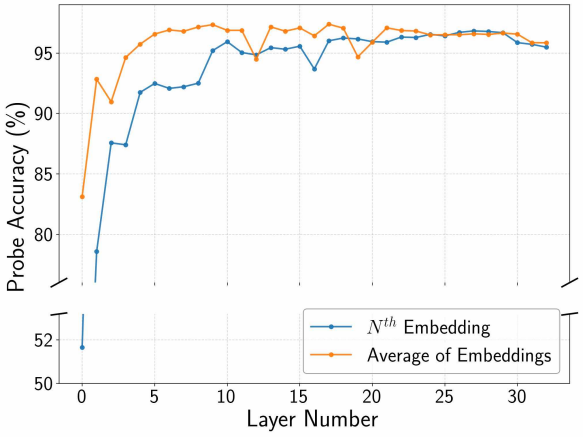

Figure G.39: **Democracy** probe accuracy for `Llama-3-8B` using average and N^{th} embeddings vs. layer

Probed LLM	Probed Layer	# Probe parameters 20	40	80	max
`Llama-3-8B`	1	71	73	78	83
	15	61	73	79	90
	32	77	83	87	92
`Gemma-2-2B`	1	68	73	77	85
	15	67	76	82	88
	26	73	81	86	89
`Qwen2.5-0.5B`	1	68	72	75	74
	15	64	77	83	89
	24	73	80	83	90

- All results are in percentage (%).
- "max" denotes 4,096 for `Llama-3-8B`, 2,304 for `Gemma-2-2B`, and 896 for `Qwen2.5-0.5B`.
- standard deviation for each result $\leq 2\%$.

Table G.1: **Investigation** probe accuracy across model families, sizes, layers, and probe sizes

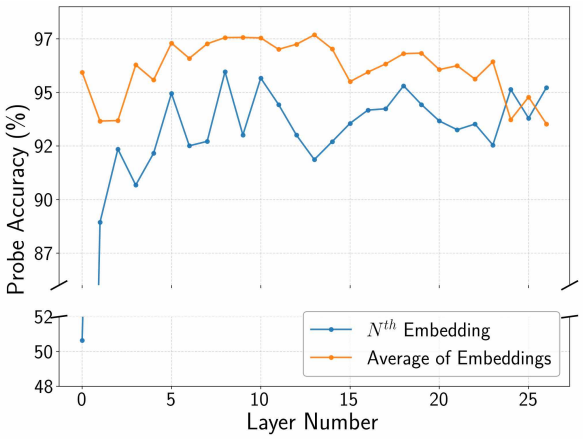

Figure G.40: **Democracy** probe accuracy for `Gemma-2-2B` using average and N^{th} embeddings vs. layer

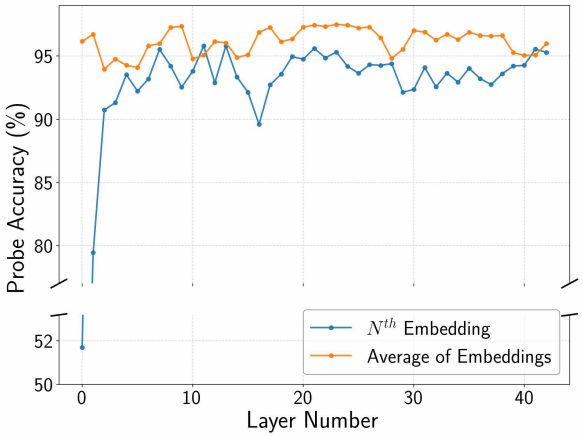

Figure G.41: **Democracy** probe accuracy for `Gemma-2-9B` using average and N^{th} embeddings vs. layer

G.4 Extended Results for Inference of Democracy

G.4.1 Probing for Democracy using Nth Embedding vs. Average Embedding

Figures G.39–G.45 illustrate the **Democracy** probe accuracies across layers of all LLMs using both the N^{th} embedding and the average of all embeddings in the respective layer.

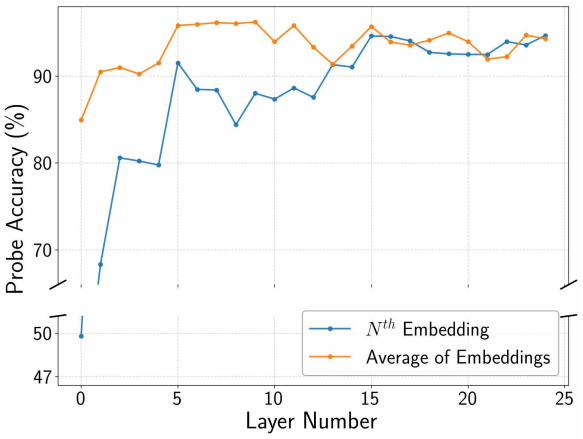

Figure G.42: **Democracy** probe accuracy for `Qwen2.5-0.5B` using average and N^{th} embeddings vs. layer

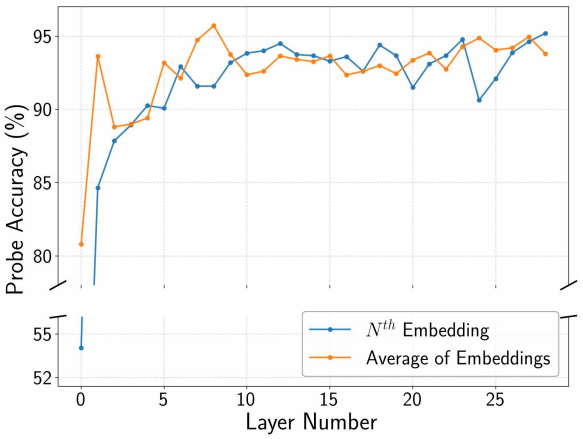

Figure G.43: **Democracy** probe accuracy for `Qwen2.5-1.5B` using average and N^{th} embeddings vs. layer

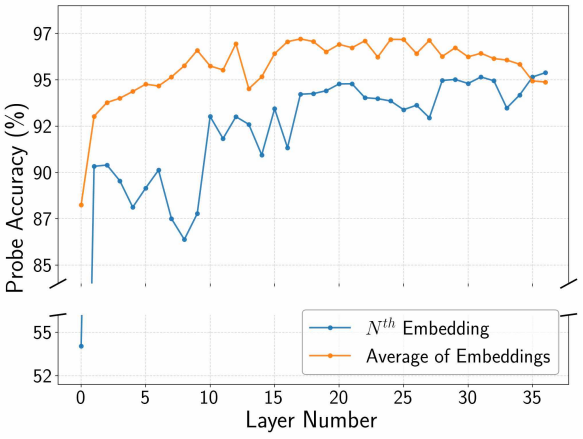

Figure G.44: **Democracy** probe accuracy for `Qwen2.5-3B` using average and N^{th} embeddings vs. layer

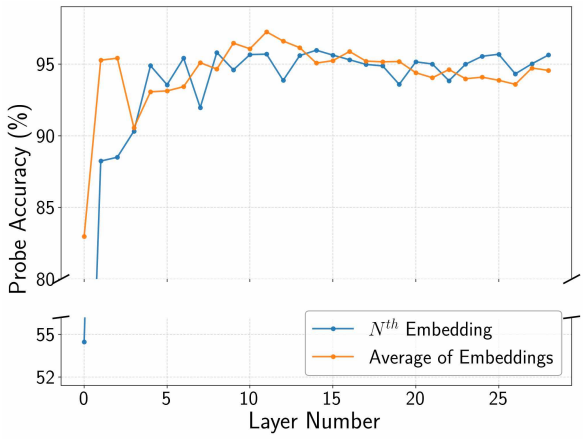

Figure G.45: **Democracy** probe accuracy for `Qwen2.5-7B` using average and N^{th} embeddings vs. layer

G.4.2 Democracy Probe Cross-Check

Figures G.46, G.47, and G.48 show the **Democracy** probe accuracy versus probe size for `Llama-3-8B`, `Gemma-2-2B`, and `Qwen2.5-0.5B`, respectively, and Table G.2 shows a summary of these results. Figures G.49–G.55 show the probe accuracies across layers for all LLMs when the probes are trained on the control task (randomizing embeddings or labels).

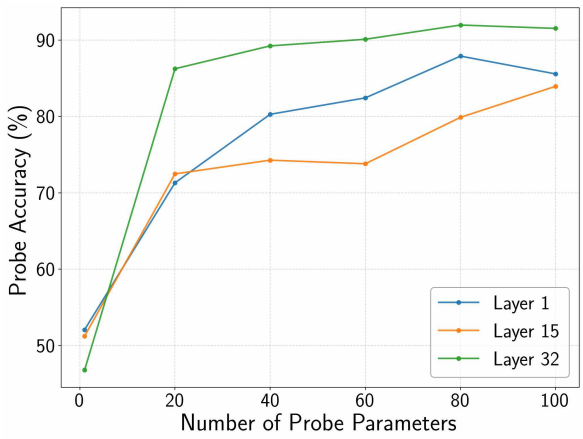

Figure G.46: **Democracy** probe accuracy for `Llama-3-8B` as a function of probe size

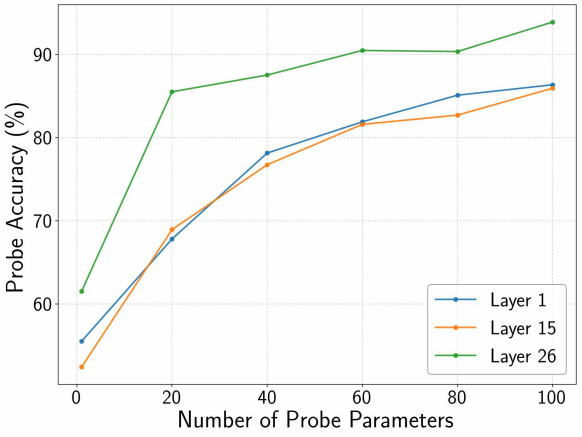

Figure G.47: **Democracy** probe accuracy for `Gemma-2-2B` as a function of probe size

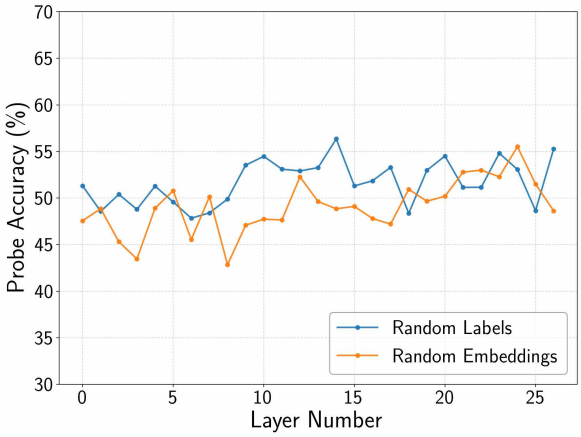

Figure G.50: **Democracy** probe accuracy across layers in `Gemma-2-2B` using random embeddings or random labels during probe training

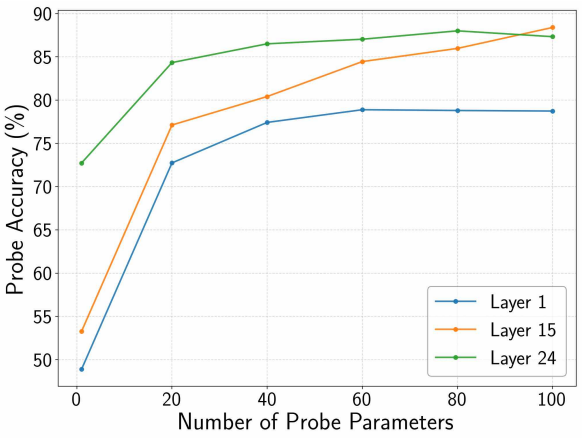

Figure G.48: **Democracy** probe accuracy for `Qwen2.5-0.5B` as a function of probe size

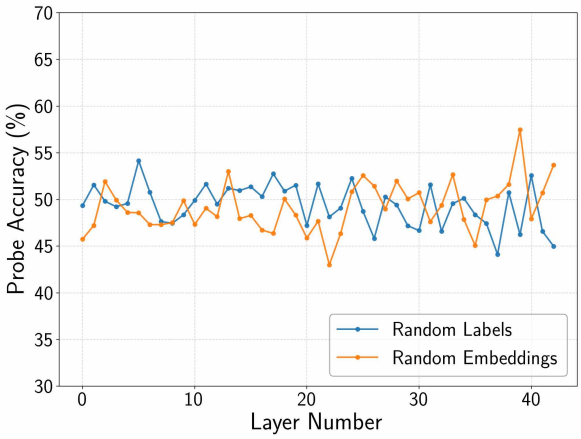

Figure G.51: **Democracy** probe accuracy across layers in `Gemma-2-9B` using random embeddings or random labels during probe training

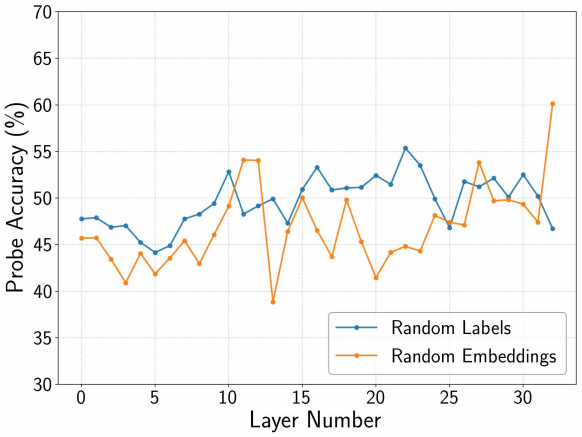

Figure G.49: **Democracy** probe accuracy across layers in `Llama-3-8B` using random embeddings or random labels during probe training

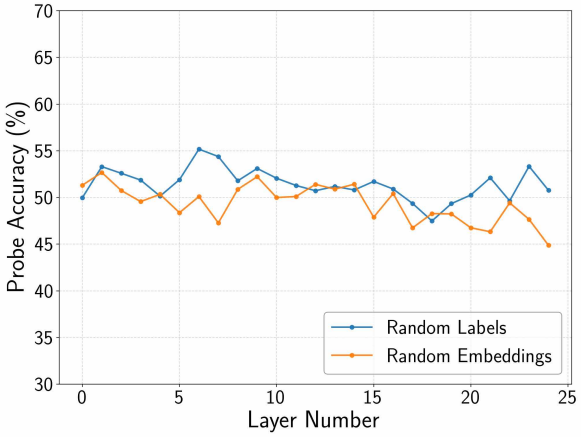

Figure G.52: **Democracy** probe accuracy across layers in `Qwen2.5-0.5B` using random embeddings or random labels during probe training

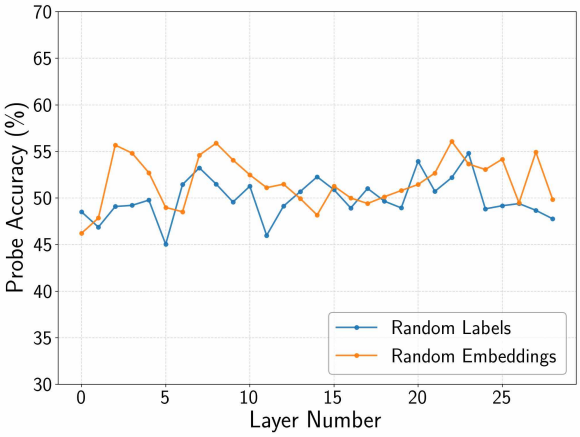

Figure G.53: **Democracy** probe accuracy across layers in `Qwen2.5-1.5B` using random embeddings or random labels during probe training

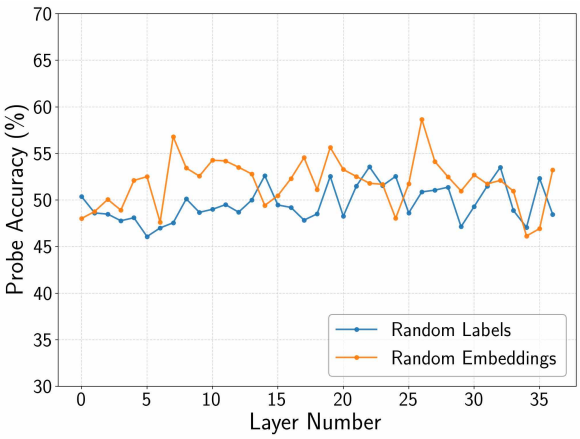

Figure G.54: **Democracy** probe accuracy across layers in `Qwen2.5-3B` using random embeddings or random labels during probe training

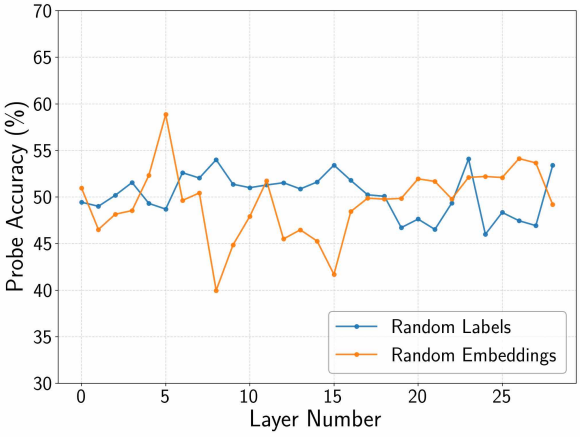

Figure G.55: **Democracy** probe accuracy across layers in `Qwen2.5-7B` using random embeddings or random labels during probe training

Probed LLM	Probed Layer	# Probe parameters 20	40	80	max
`Llama-3-8B`	1	71	80	88	79
	15	72	74	80	96
	32	86	89	92	95
`Gemma-2-2B`	1	68	78	85	89
	15	69	77	83	94
	26	86	88	90	95
`Qwen2.5-0.5B`	1	73	77	79	68
	15	77	80	86	95
	24	84	87	88	95

- All results are in percentage (%).
- "max" denotes 4,096 for `Llama-3-8B`, 2,304 for `Gemma-2-2B`, and 896 for `Qwen2.5-0.5B`.
- standard deviation for each result $\leq 3\%$.

Table G.2: **Democracy** probe accuracy across model families, sizes, layers, and probe sizes

G.5 Extended Results for Inference of Envy

G.5.1 Probing for Envy using Nth Embedding vs. Average Embedding

Figures G.56–G.62 illustrate the **Envy** probe accuracies across layers of all LLMs using both the N^{th} embedding and the average of all embeddings in the respective layer.

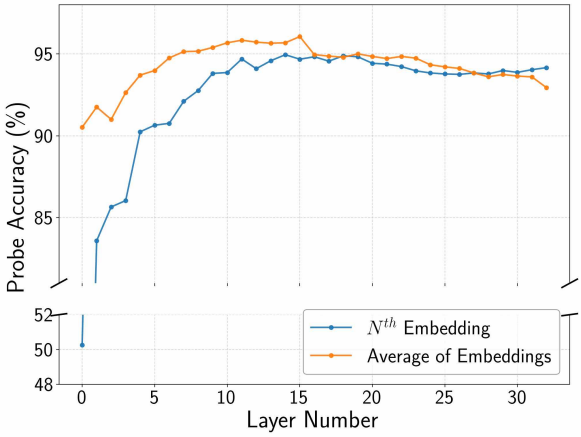

Figure G.56: **Envy** probe accuracy for `Llama-3-8B` using average and N^{th} embeddings vs. layer

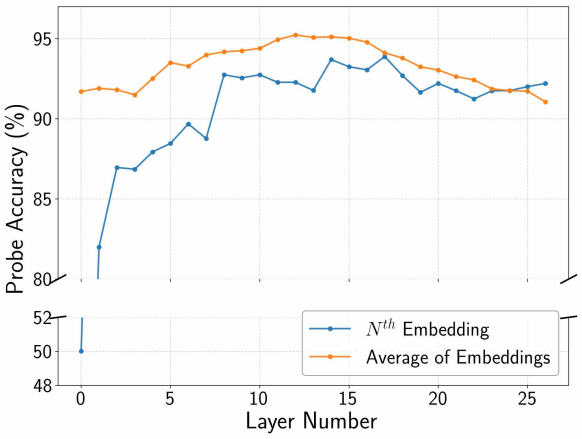

Figure G.57: **Envy** probe accuracy for `Gemma-2-2B` using average and N^{th} embeddings vs. layer

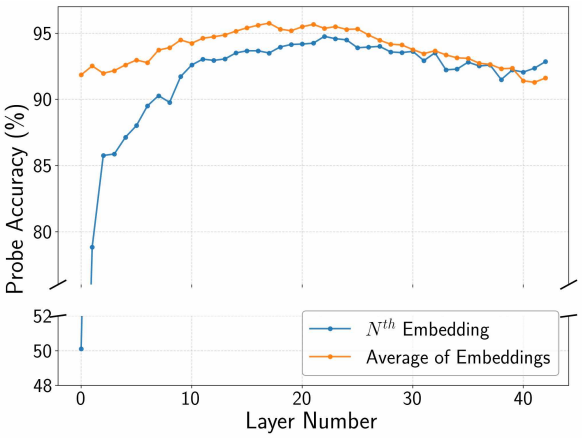

Figure G.58: **Envy** probe accuracy for `Gemma-2-9B` using average and N^{th} embeddings vs. layer

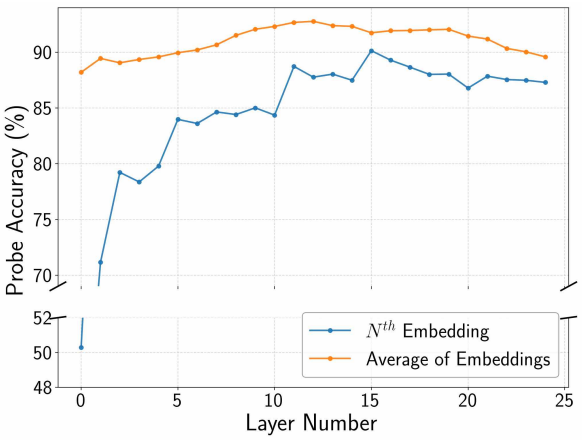

Figure G.59: **Envy** probe accuracy for `Qwen2.5-0.5B` using average and N^{th} embeddings vs. layer

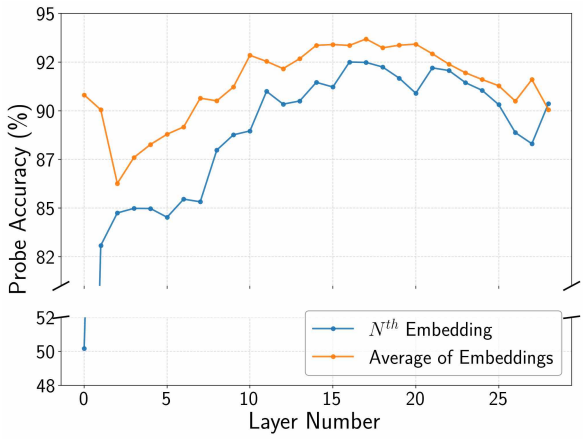

Figure G.60: **Envy** probe accuracy for `Qwen2.5-1.5B` using average and N^{th} embeddings vs. layer

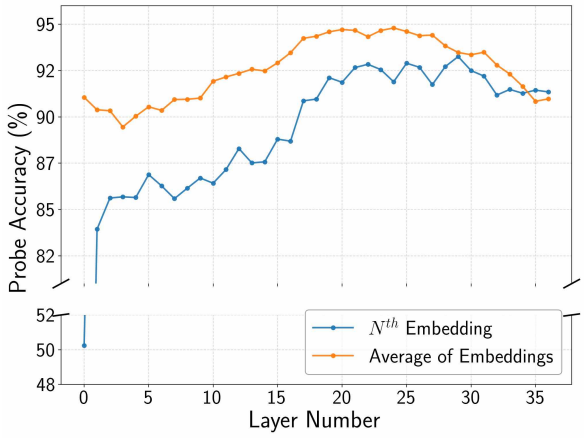

Figure G.61: **Envy** probe accuracy for `Qwen2.5-3B` using average and N^{th} embeddings vs. layer

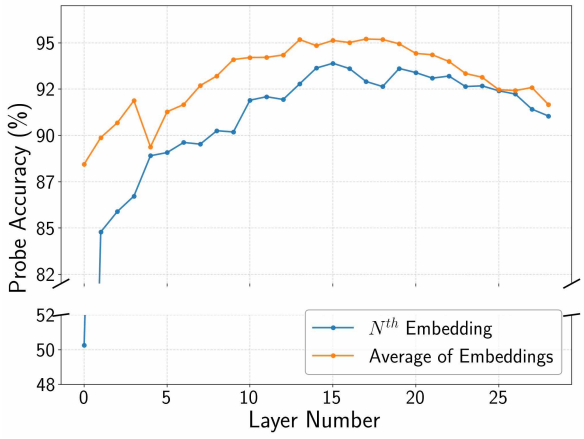

Figure G.62: **Envy** probe accuracy for `Qwen2.5-7B` using average and N^{th} embeddings vs. layer

G.5.2 Envy Probe Cross-Check

Figures G.63, G.64, and G.65 show the **Envy** probe accuracy versus probe size for `Llama-3-8B`,

Gemma-2-2B, and Qwen2.5-0.5B, respectively, and Table G.3 shows a summary of these results. Figures G.66–G.72 show the probe accuracies across layers for all LLMs when the probes are trained on the control task (randomizing embeddings or labels).

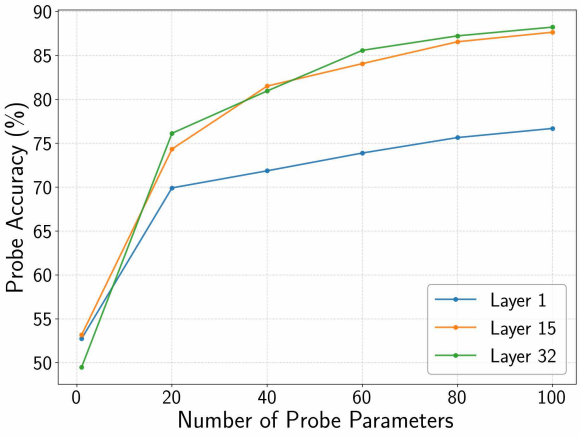

Figure G.63: **Envy** probe accuracy for Llama-3-8B as a function of probe size

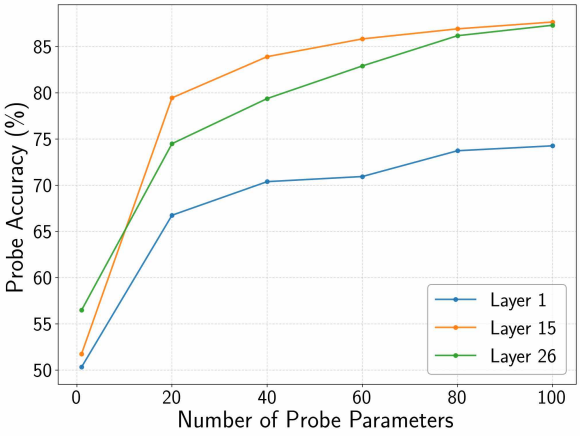

Figure G.64: **Envy** probe accuracy for Gemma-2-2B as a function of probe size

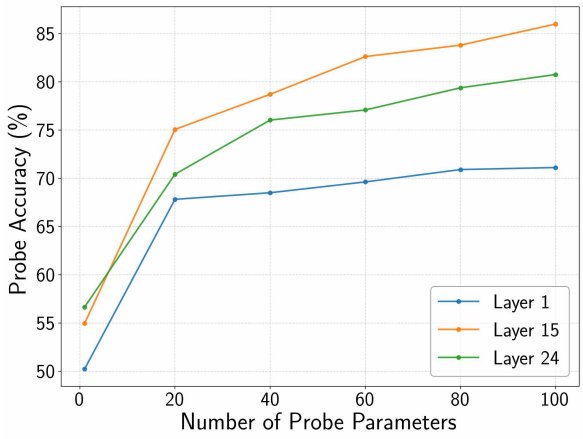

Figure G.65: **Envy** probe accuracy for Qwen2.5-0.5B as a function of probe size

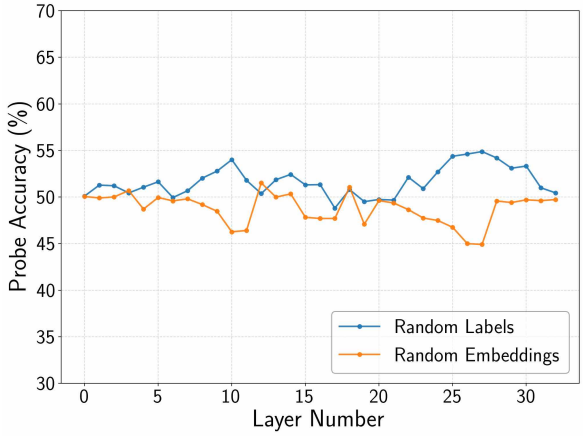

Figure G.66: **Envy** probe accuracy across layers in Llama-3-8B using random embeddings or random labels during probe training

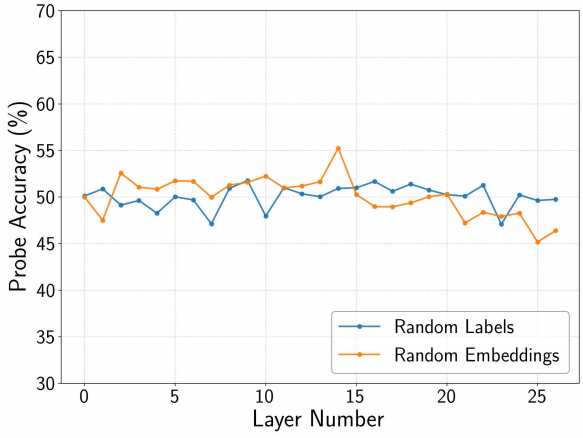

Figure G.67: **Envy** probe accuracy across layers in Gemma-2-2B using random embeddings or random labels during probe training

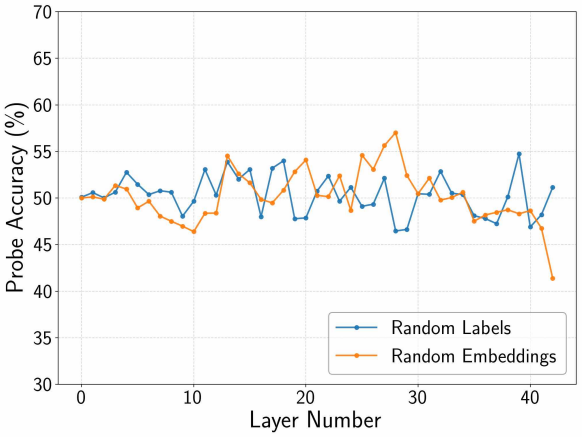

Figure G.68: **Envy** probe accuracy across layers in `Gemma-2-9B` using random embeddings or random labels during probe training

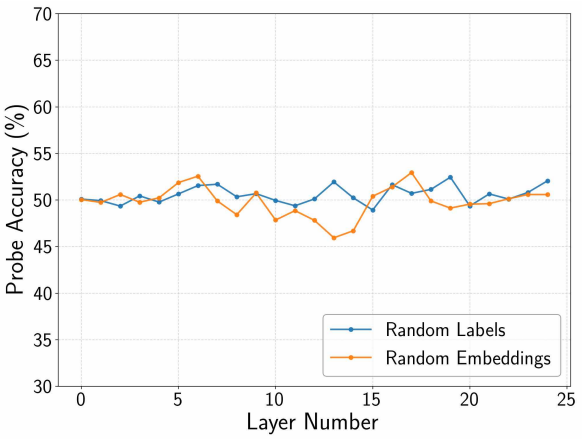

Figure G.69: **Envy** probe accuracy across layers in `Qwen2.5-0.5B` using random embeddings or random labels during probe training

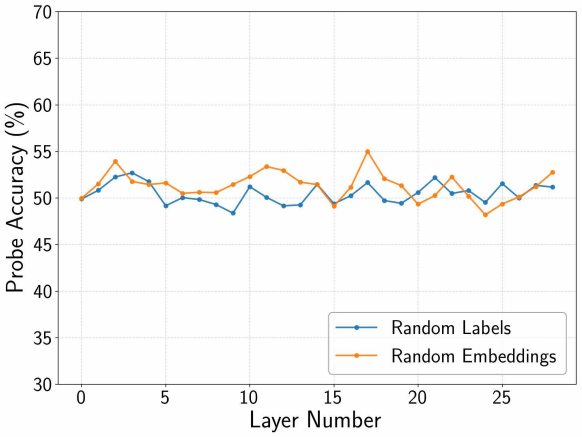

Figure G.70: **Envy** probe accuracy across layers in `Qwen2.5-1.5B` using random embeddings or random labels during probe training

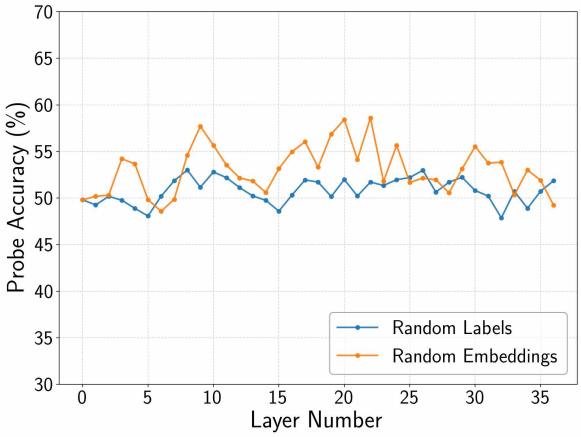

Figure G.71: **Envy** probe accuracy across layers in `Qwen2.5-3B` using random embeddings or random labels during probe training

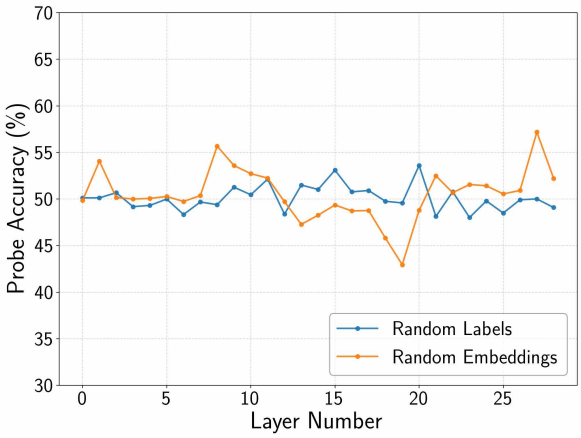

Figure G.72: **Envy** probe accuracy across layers in `Qwen2.5-7B` using random embeddings or random labels during probe training

Probed LLM	Probed Layer	# Probe parameters 20	40	80	max
`Llama-3-8B`	1	70	72	76	84
	15	74	82	87	95
	32	76	81	87	94
`Gemma-2-2B`	1	67	70	74	82
	15	79	84	87	93
	26	74	79	86	92
`Qwen2.5-0.5B`	1	68	69	71	71
	15	75	79	84	90
	24	70	76	79	87

- All results are in percentage (%).
- "max" denotes 4,096 for `Llama-3-8B`, 2,304 for `Gemma-2-2B`, and 896 for `Qwen2.5-0.5B`.
- standard deviation for each result $\leq 1\%$.

Table G.3: **Envy** probe accuracy across model families, sizes, layers, and probe sizes

H Extended Results for Waxing and Waning of Concepts

In addition to investigating whether concepts wax and wane in an LLM's embeddings as its context expands, we also studied how this behavior varies across layers to identify which layer best captures this change. For each layer, we generated kernel density estimation (KDE) plots showing the distribution of the probe's output values for the target concept across different story segments (e.g., paragraph 1, transition sentence 1, paragraph 2), aggregated over all stories in the concept dataset. Figure H.1 shows an example for the **ambition** probe at layer 13 of `Llama-3-8B` (using the final subword token embedding), with KDEs for the transition sentences shown in shades of green and those for paragraphs in shades of red.

A layer that accurately captures waxing and waning should show the transition KDEs concentrated above 0.5 and paragraph KDEs below 0.5. In contrast, a layer that does not capture the waxing and waning would either have all KDEs for paragraphs and transitions concentrated around the same point, as shown in Figure H.2 (using cumulative mean embeddings), or have the KDEs for each segment widely distributed.

We stacked these plots for all LLM layers to identify the one that best tracks the waxing and waning, as shown ahead. Table H.1 shows the layer in each LLM that best captures the waxing and waning for each investigated concept.

We use the probes for the corresponding layers to track the model embedding's word-level behavior. Using the story datasets, we obtained an aggregate view of each probe's outputs. This was computed as the average probe output for each word position across all 50 stories. For example, the aggregate value for word number 3 in sentence 15 is determined by the average of the corresponding probe outputs for all the words in that position across all stories. To obtain correct average values, all sentences at the same sentence position across stories must have the same number of words, which might not be the case. To alleviate this issue, we applied left-padding to each sentence to align its length with longest sentence at that position. The padded tokens were excluded from the averaging computation.

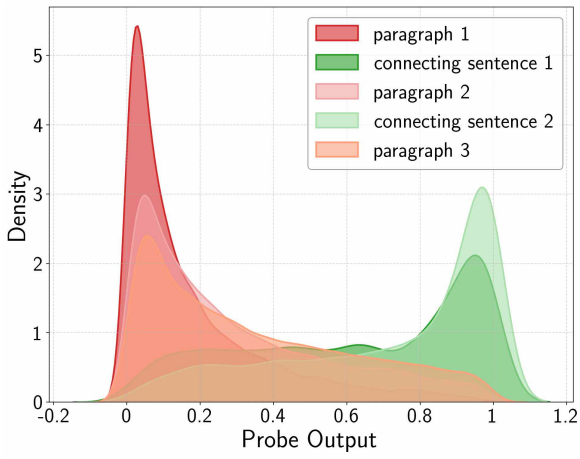

Figure H.1: Kernel density estimation of the **Ambition** probe's output values across story segments, aggregated over all stories, from layer 13 of `Llama-3-8B` (using final subword token embeddings)

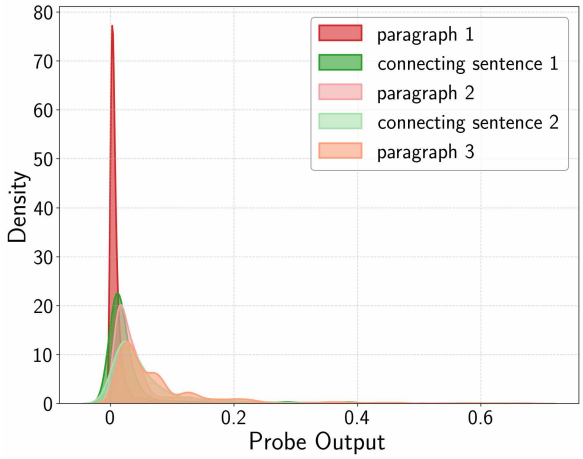

Figure H.2: Kernel density estimation of the **Ambition** probe's output values across story segments, aggregated over all stories, from layer 13 of `Llama-3-8B` (using cumulative mean embeddings)

Probed LLM	Layer that best captures concept waxing and waning			
	Ambition	Investigation	Democracy	Envy
`Llama-3-8B`	13	31	7	10
`Gemma-2-2B`	12	6	16	14
`Gemma-2-9B`	22	11	11	19
`Qwen2.5-0.5B`	20	16	5	15
`Qwen2.5-1.5B`	15	18	14	20
`Qwen2.5-3B`	27	25	14	27
`Qwen2.5-7B`	19	12	12	24

Table H.1: LLM layers that best capture the waxing and waning of the investigated concepts

H.1 Tracking Waxing and Waning of Ambition

H.1.1 Layer-Wise KDEs for Ambition Probe Outputs

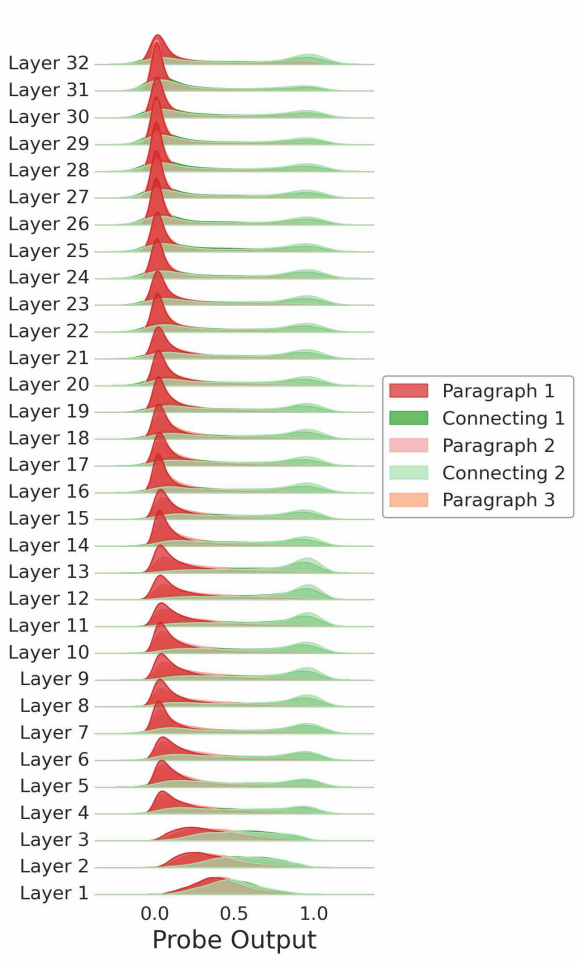

Figure H.3: Layer-wise KDEs for **ambition** probe outputs in `Llama-3-8B`

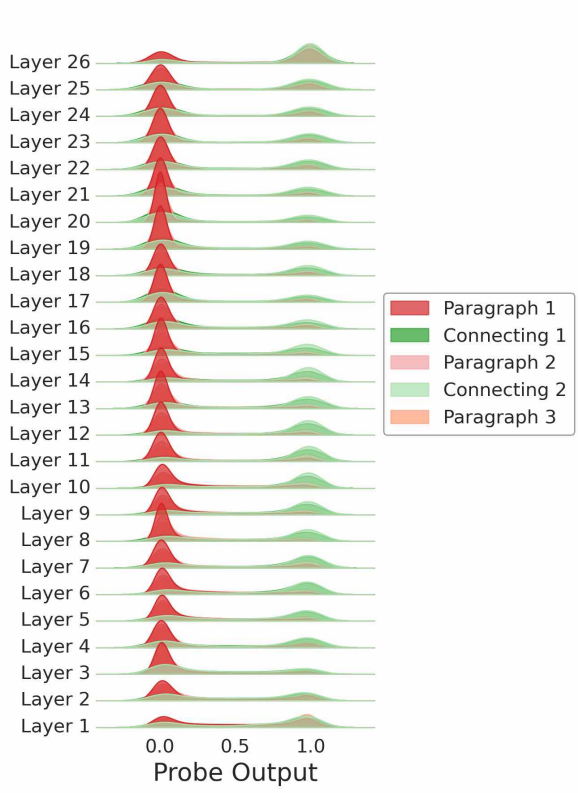

Figure H.4: Layer-wise KDEs for **ambition** probe outputs in `Gemma-2-2B`

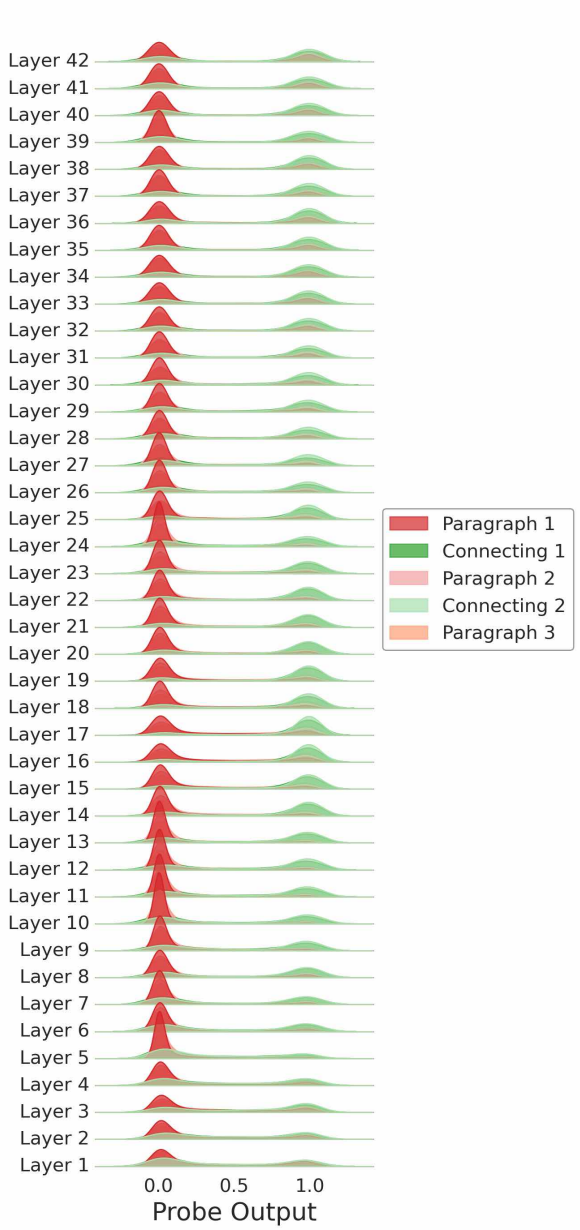

Figure H.5: Layer-wise KDEs for **ambition** probe outputs in `Gemma-2-9B`

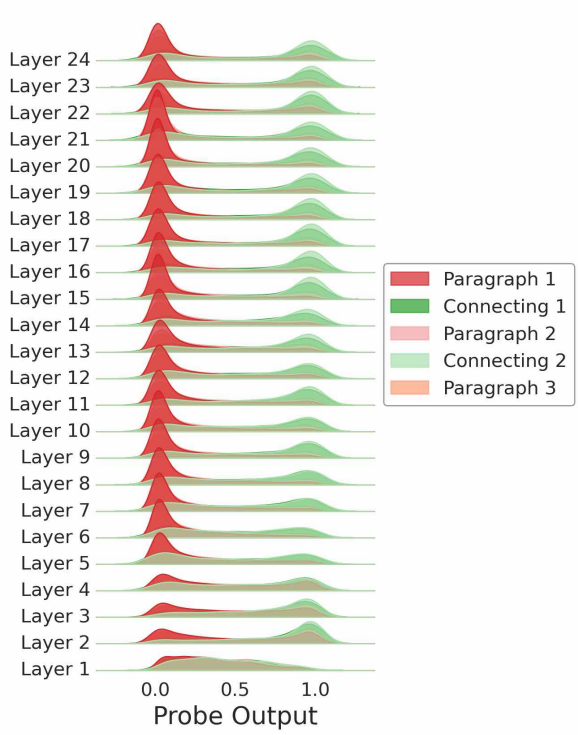

Figure H.6: Layer-wise KDEs for **ambition** probe outputs in `Qwen2.5-0.5B`

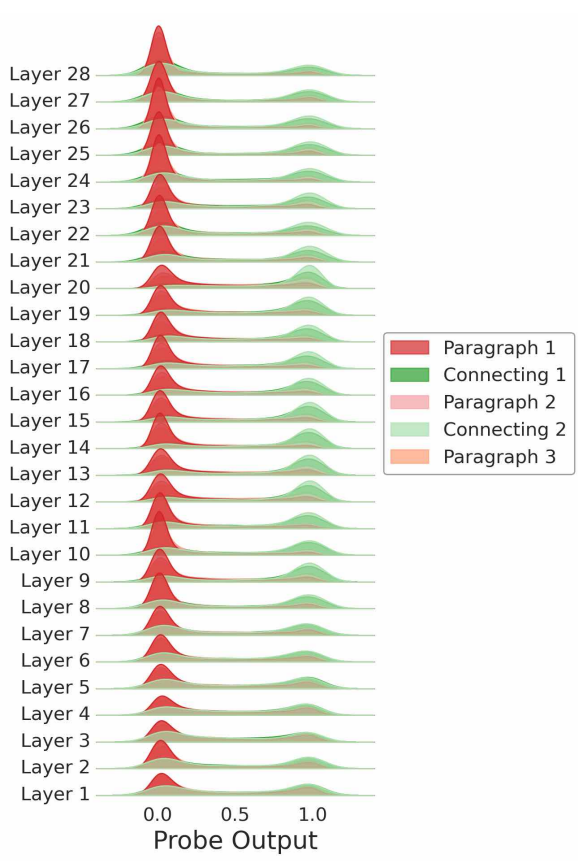

Figure H.7: Layer-wise KDEs for **ambition** probe outputs in `Qwen2.5-1.5B`

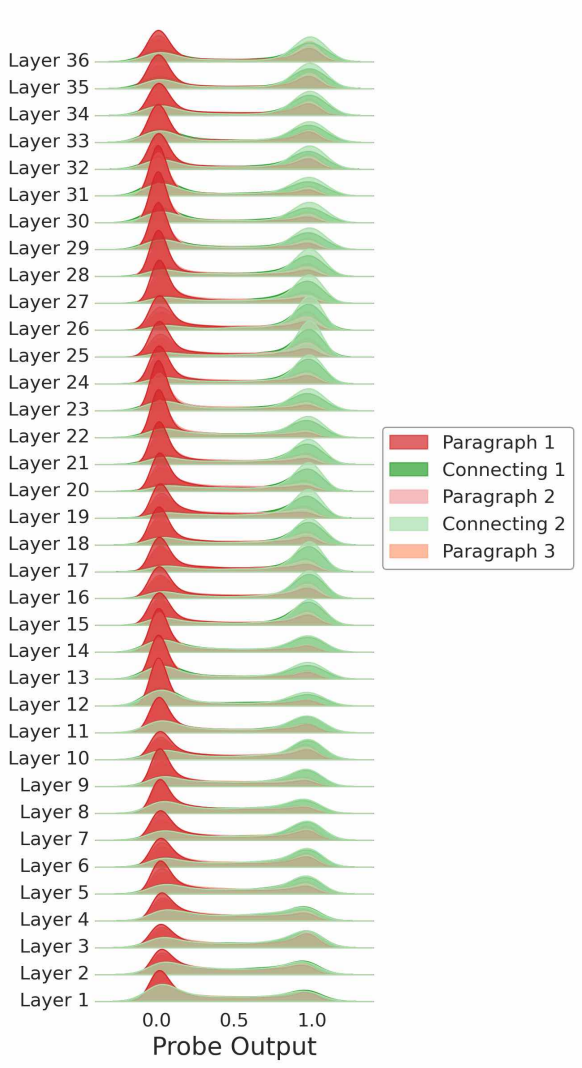

Figure H.8: Layer-wise KDEs for **ambition** probe outputs in `Qwen2.5-3B`

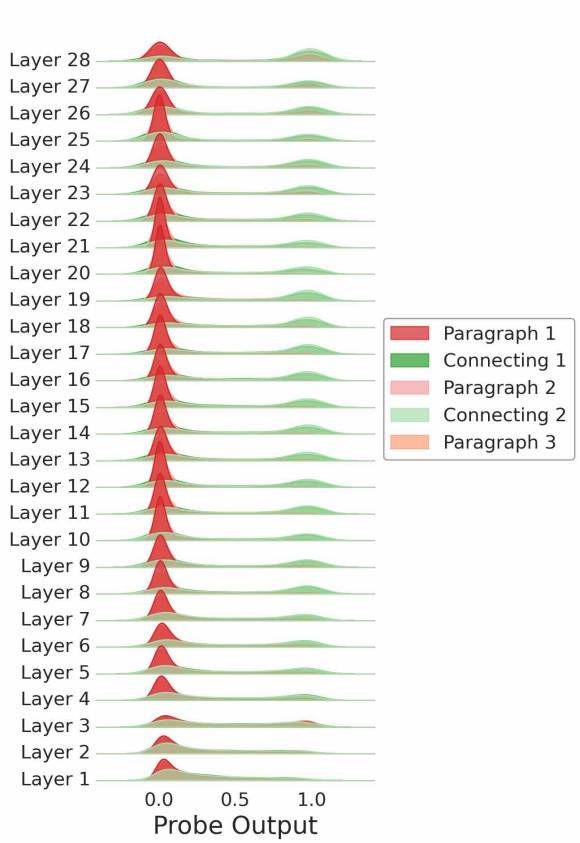

Figure H.9: Layer-wise KDEs for **ambition** probe outputs in `Qwen2.5-7B`

H.1.2 Ambition Probe Results for Best Layers

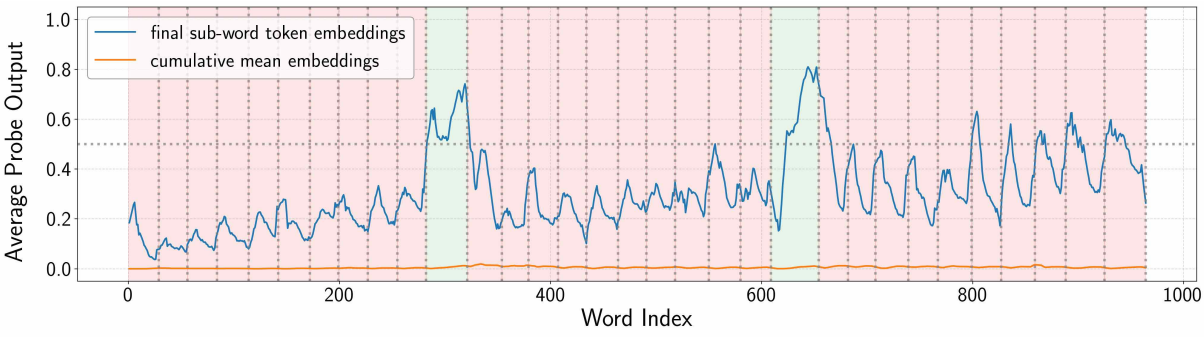

Figure H.10: **Ambition** probe outputs across words using both representative embeddings in `Gemma-2-2B`

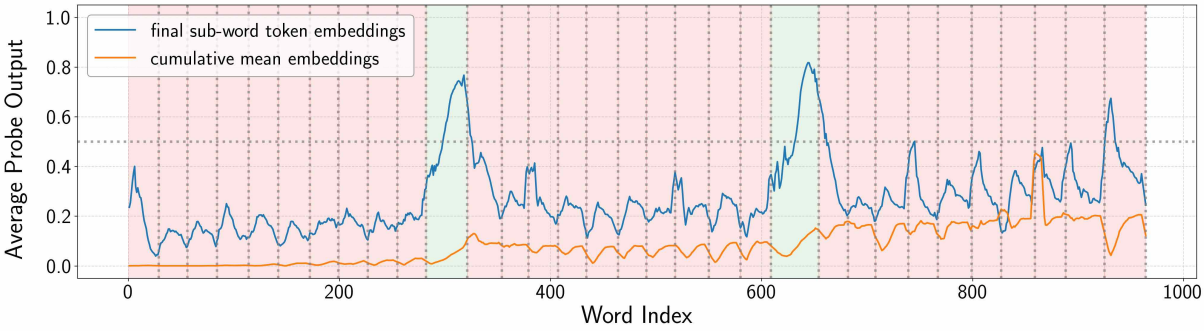

Figure H.11: **Ambition** probe outputs across words using both representative embeddings in `Gemma-2-9B`

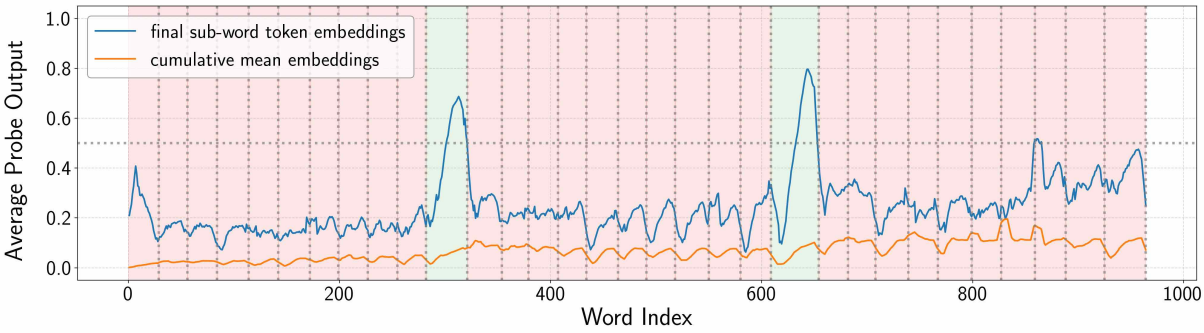

Figure H.12: **Ambition** probe outputs across words using both representative embeddings in `Qwen2.5-0.5B`

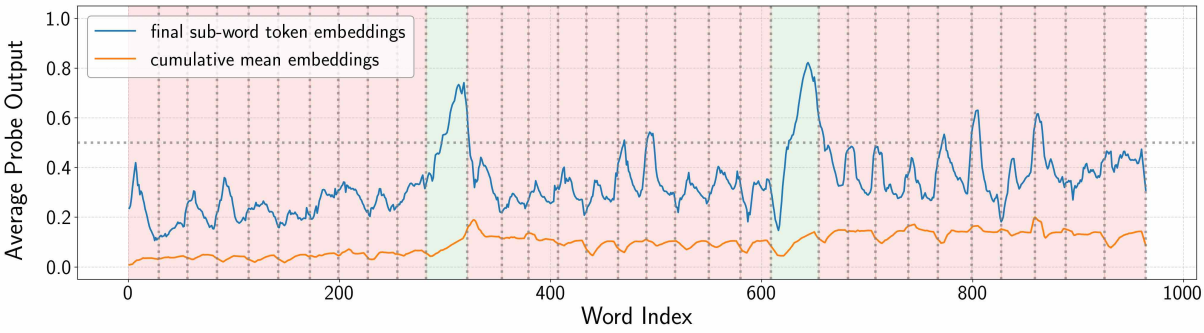

Figure H.13: **Ambition** probe outputs across words using both representative embeddings in `Qwen2.5-1.5B`

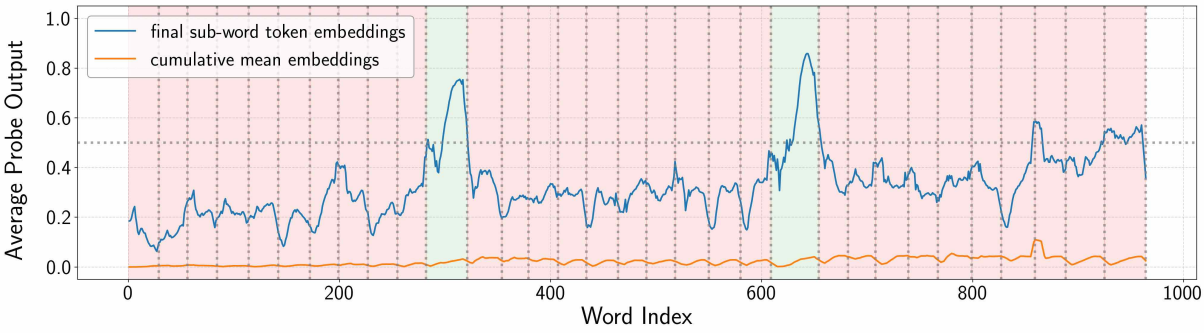

Figure H.14: **Ambition** probe outputs across words using both representative embeddings in `Qwen2.5-3B`

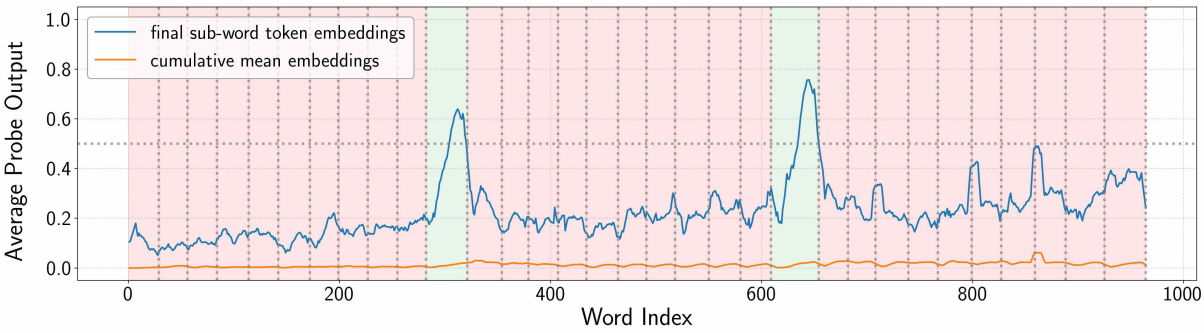

Figure H.15: **Ambition** probe outputs across words using both representative embeddings in `Qwen2.5-7B`

H.2 Tracking Waxing and Waning of Investigation

H.2.1 Layer-Wise KDEs for Investigation Probe Outputs

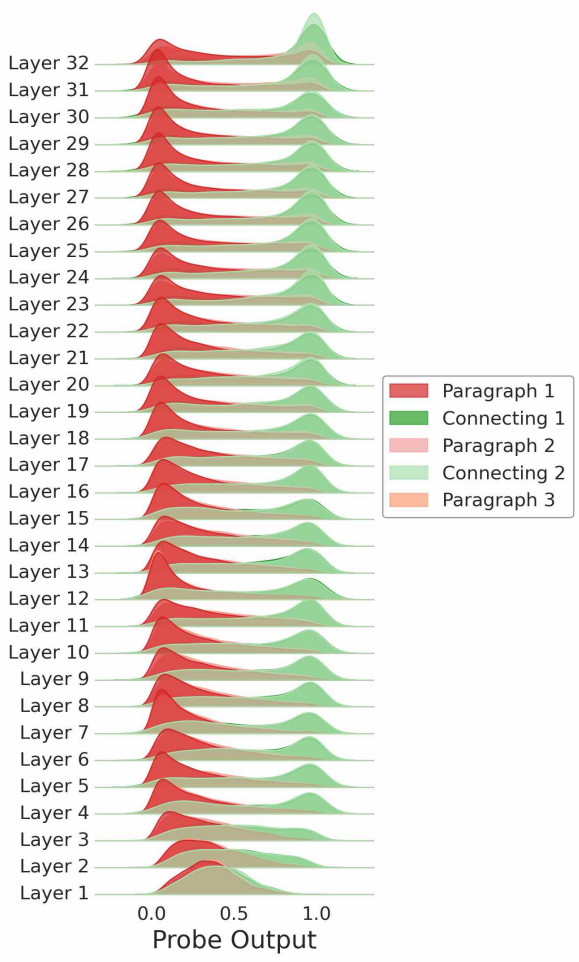

Figure H.16: Layer-wise KDEs for **investigation** probe outputs in `Llama-3-8B`

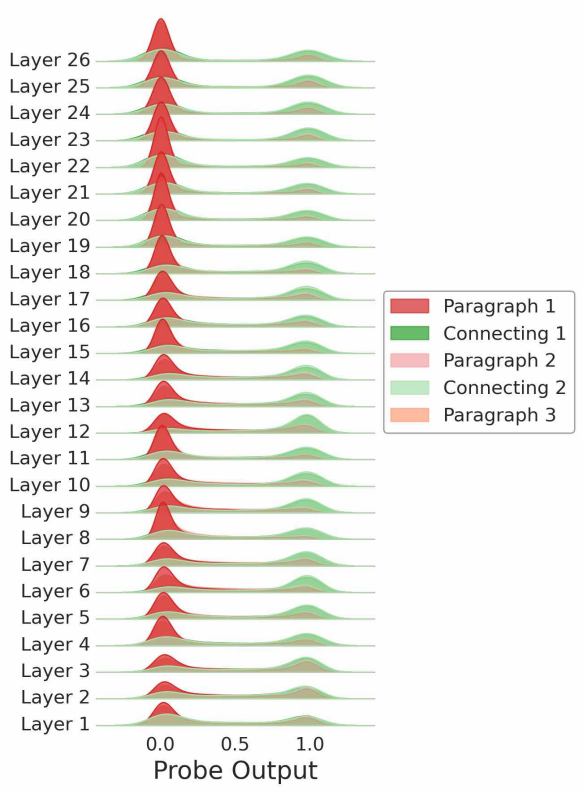

Figure H.17: Layer-wise KDEs for **investigation** probe outputs in `Gemma-2-2B`

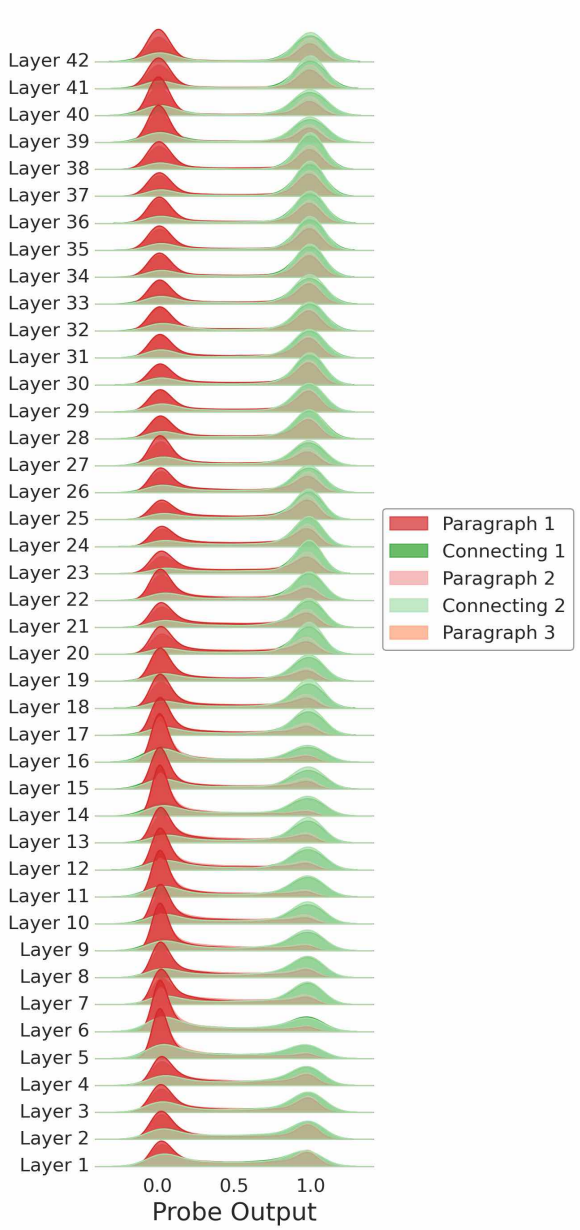

Figure H.18: Layer-wise KDEs for **investigation** probe outputs in `Gemma-2-9B`

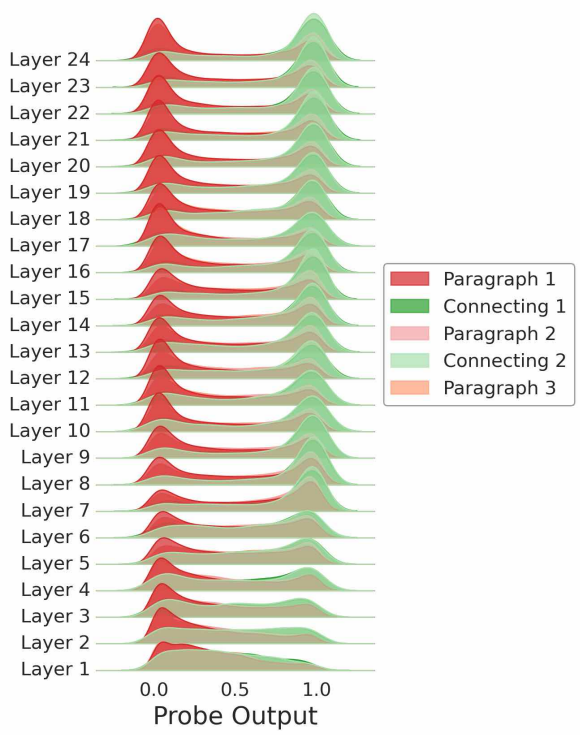

Figure H.19: Layer-wise KDEs for **investigation** probe outputs in `Qwen2.5-0.5B`

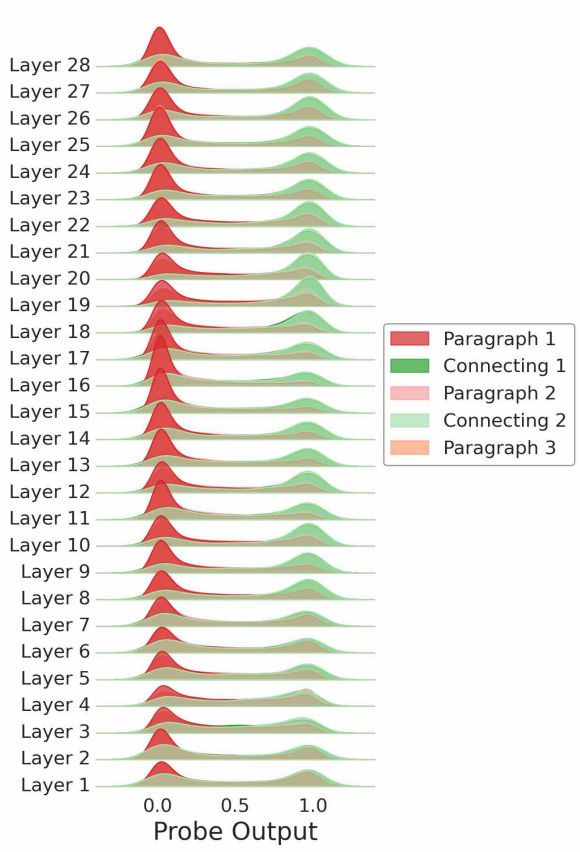

Figure H.20: Layer-wise KDEs for **investigation** probe outputs in `Qwen2.5-1.5B`

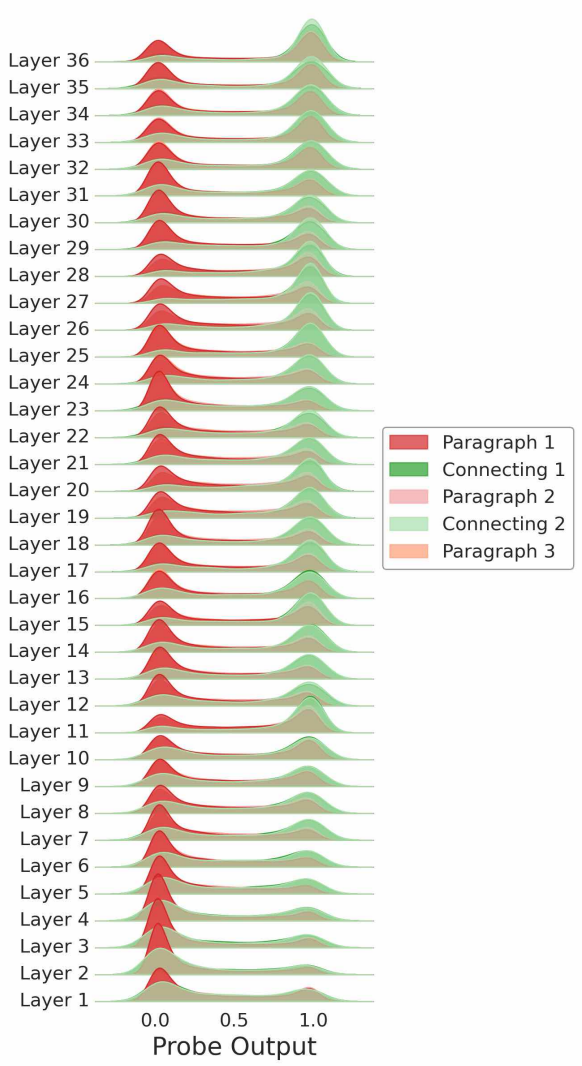

Figure H.21: Layer-wise KDEs for **investigation** probe outputs in `Qwen2.5-3B`

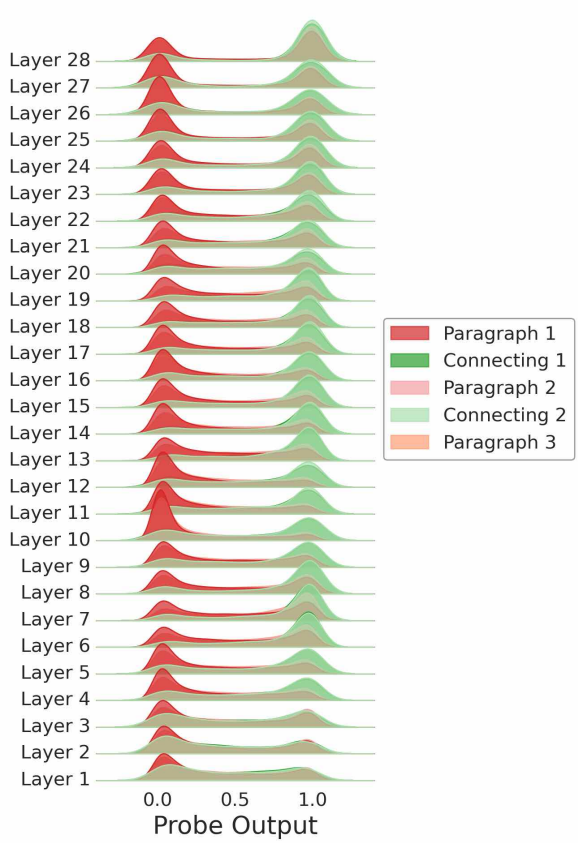

Figure H.22: Layer-wise KDEs for **investigation** probe outputs in `Qwen2.5-7B`

H.2.2 Investigation Probe Results for Best Layers

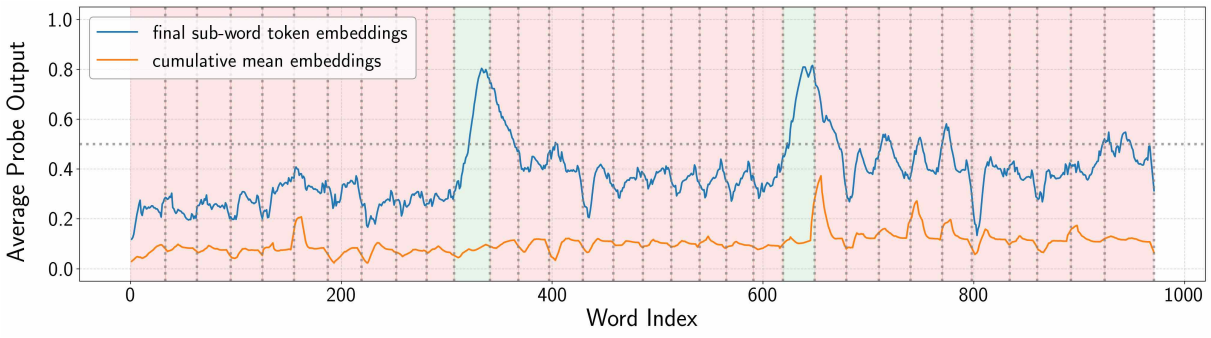

Figure H.23: **Investigation** probe outputs across words using both representative embeddings in `Llama-3-8B`

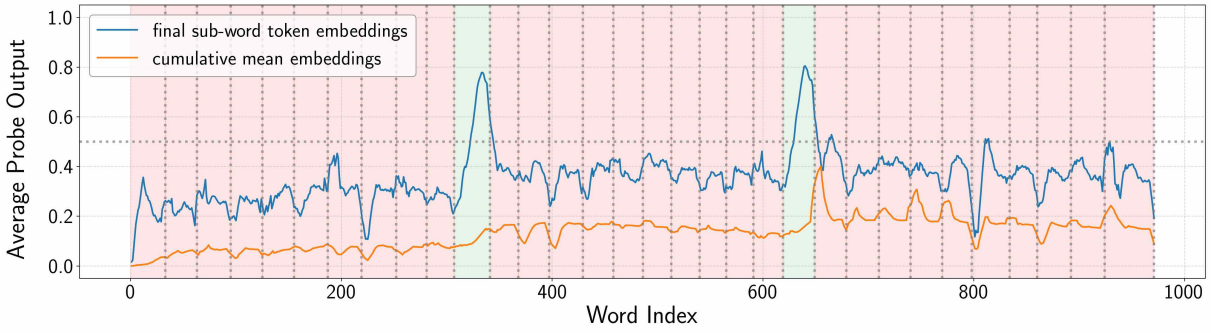

Figure H.24: **Investigation** probe outputs across words using both representative embeddings in `Gemma-2-2B`

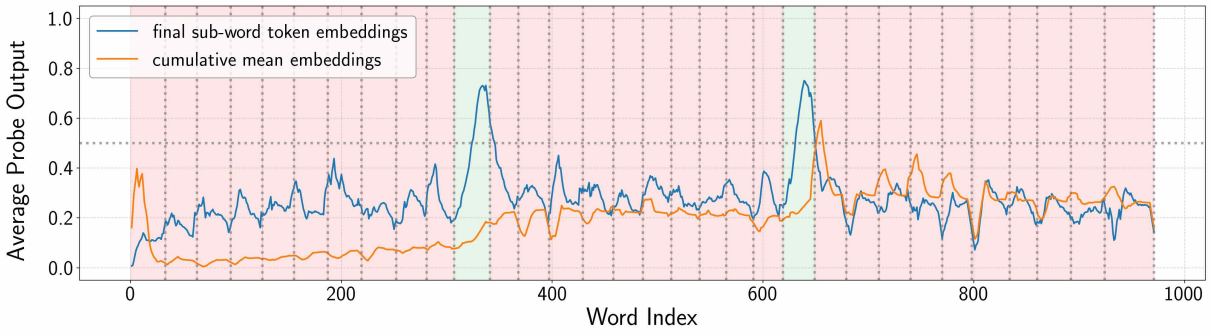

Figure H.25: **Investigation** probe outputs across words using both representative embeddings in `Gemma-2-9B`

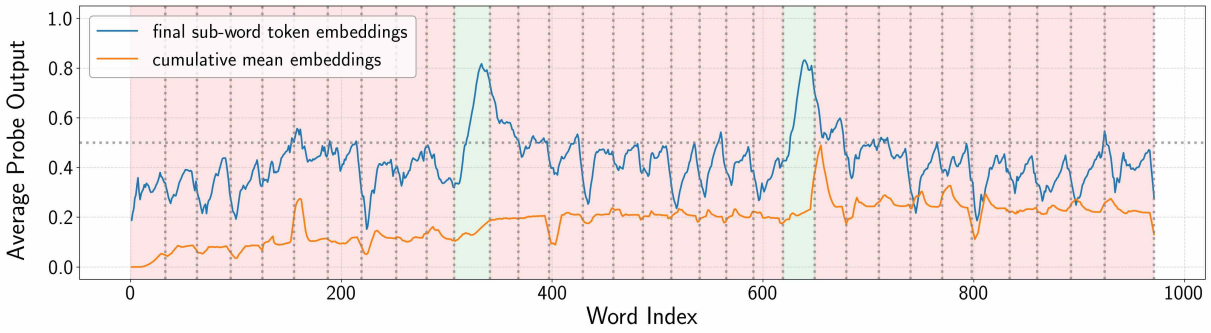

Figure H.26: **Investigation** probe outputs across words using both representative embeddings in `Qwen2.5-0.5B`

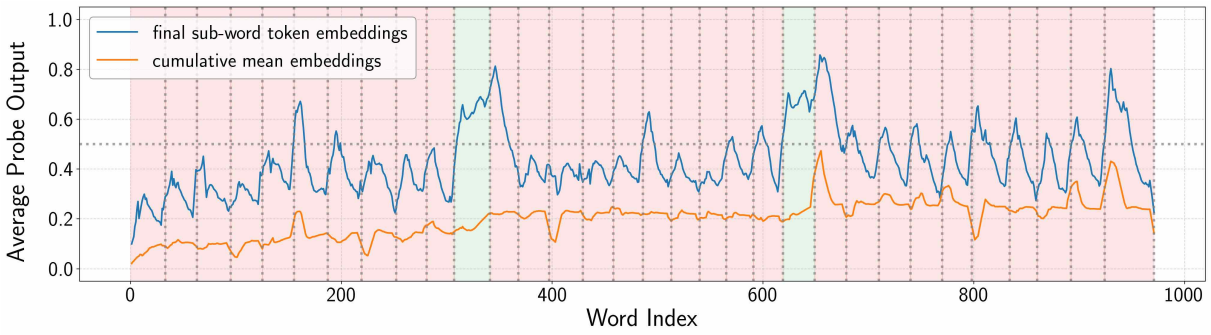

Figure H.27: **Investigation** probe outputs across words using both representative embeddings in `Qwen2.5-1.5B`

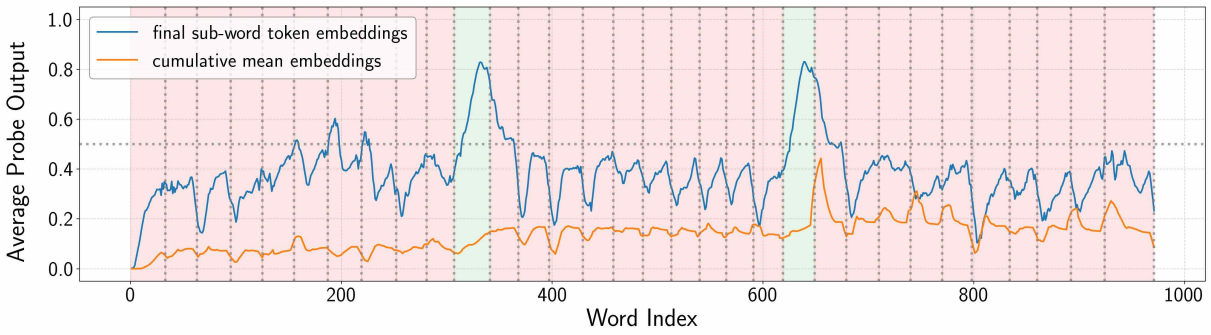

Figure H.28: **Investigation** probe outputs across words using both representative embeddings in `Qwen2.5-3B`

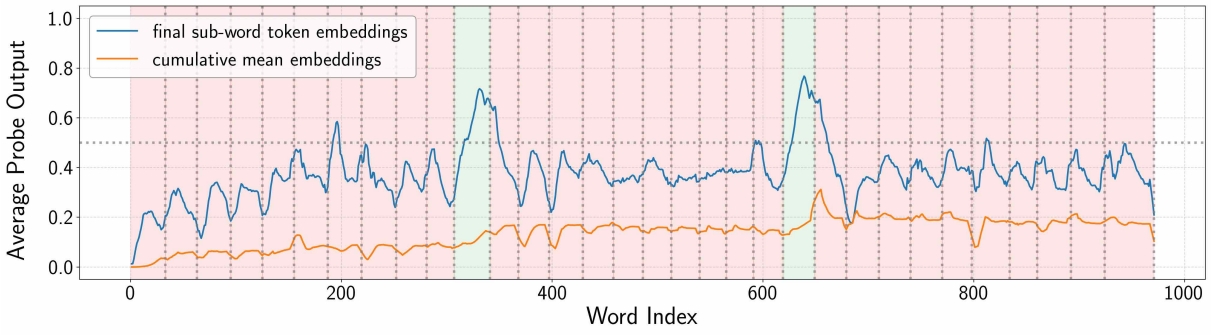

Figure H.29: **Investigation** probe outputs across words using both representative embeddings in `Qwen2.5-7B`

H.3 Tracking Waxing and Waning of Democracy

H.3.1 Layer-Wise KDEs for Democracy Probe Outputs

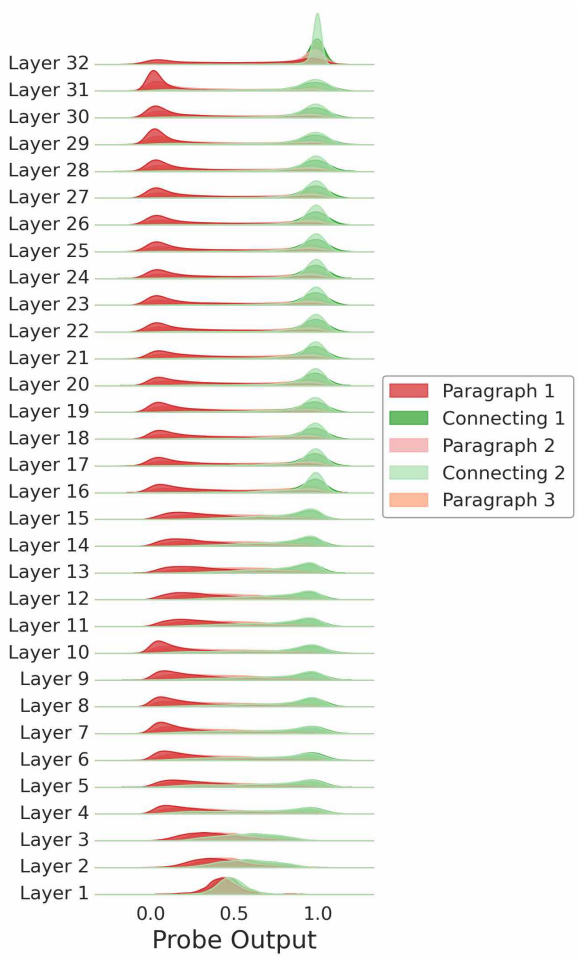

Figure H.30: Layer-wise KDEs for **democracy** probe outputs in `Llama-3-8B`

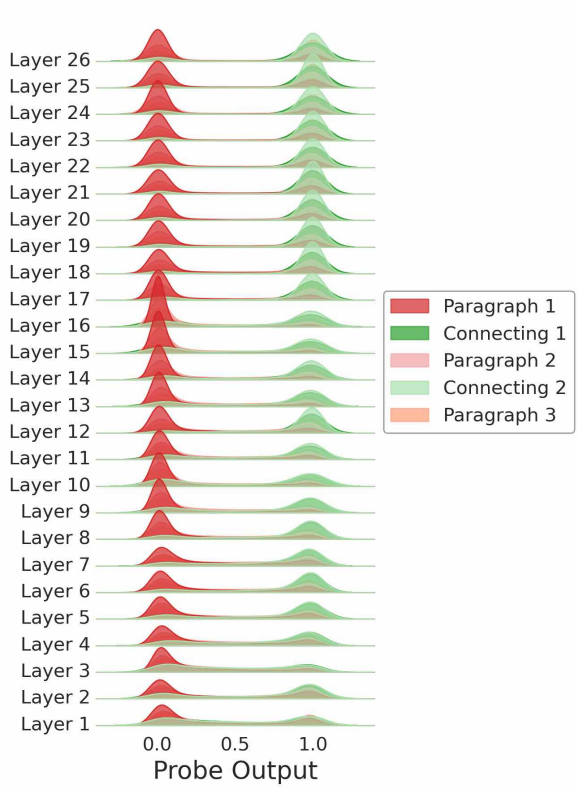

Figure H.31: Layer-wise KDEs for **democracy** probe outputs in `Gemma-2-2B`

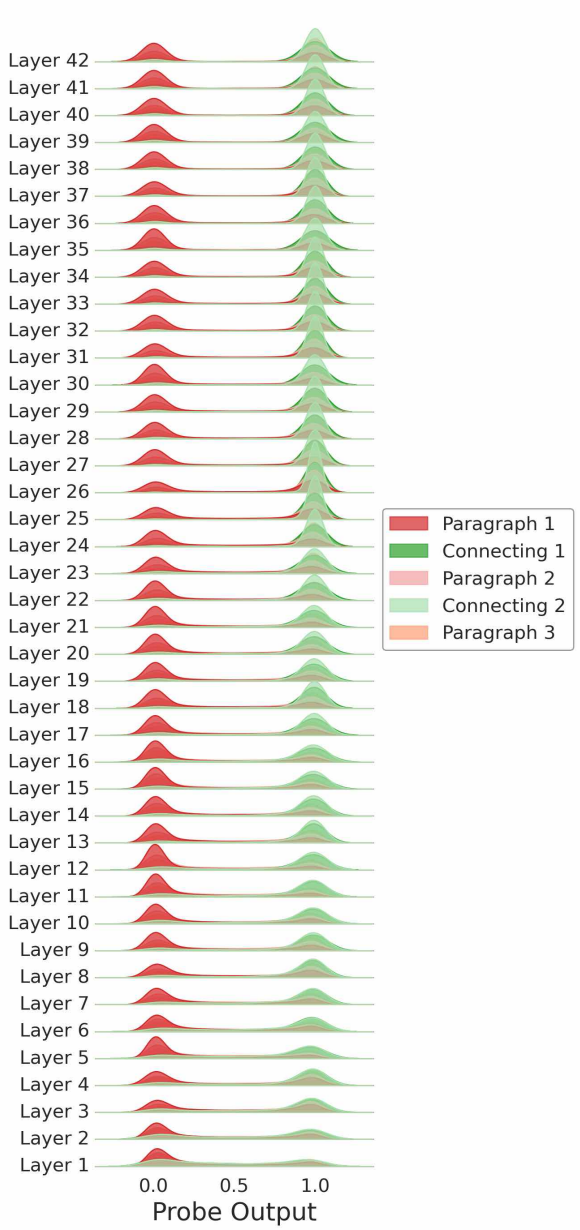

Figure H.32: Layer-wise KDEs for **democracy** probe outputs in `Gemma-2-9B`

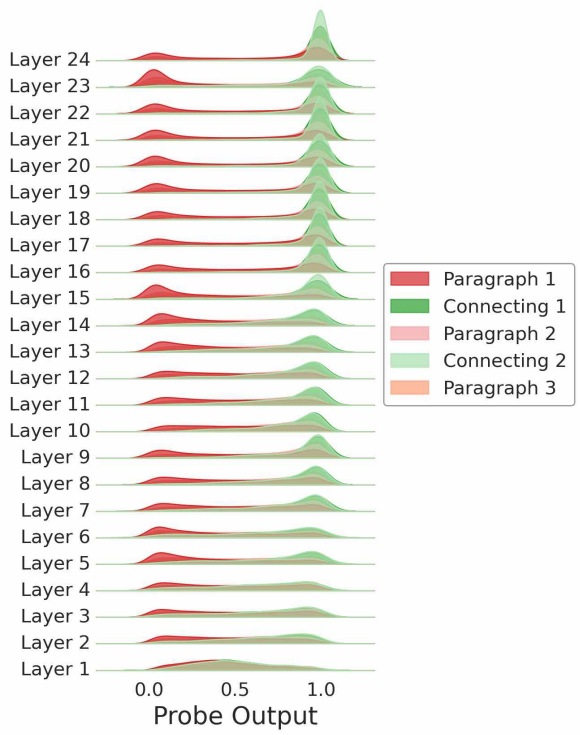

Figure H.33: Layer-wise KDEs for **democracy** probe outputs in `Qwen2.5-0.5B`

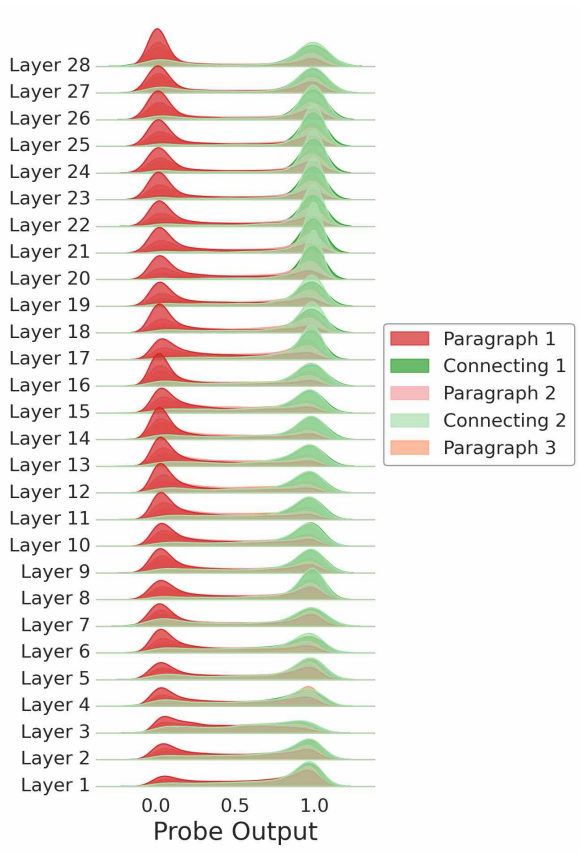

Figure H.34: Layer-wise KDEs for **democracy** probe outputs in `Qwen2.5-1.5B`

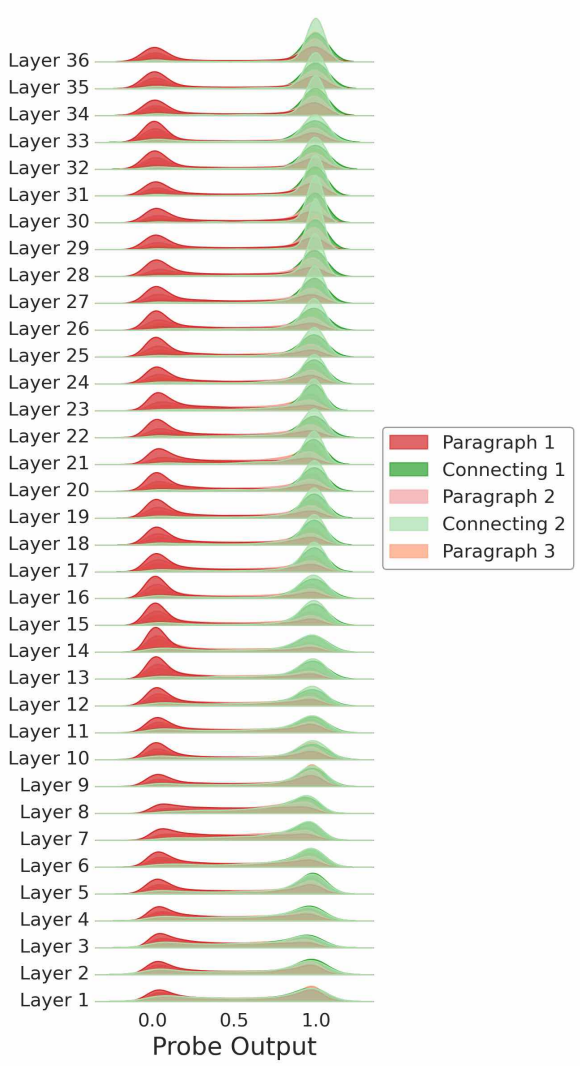

Figure H.35: Layer-wise KDEs for **democracy** probe outputs in `Qwen2.5-3B`

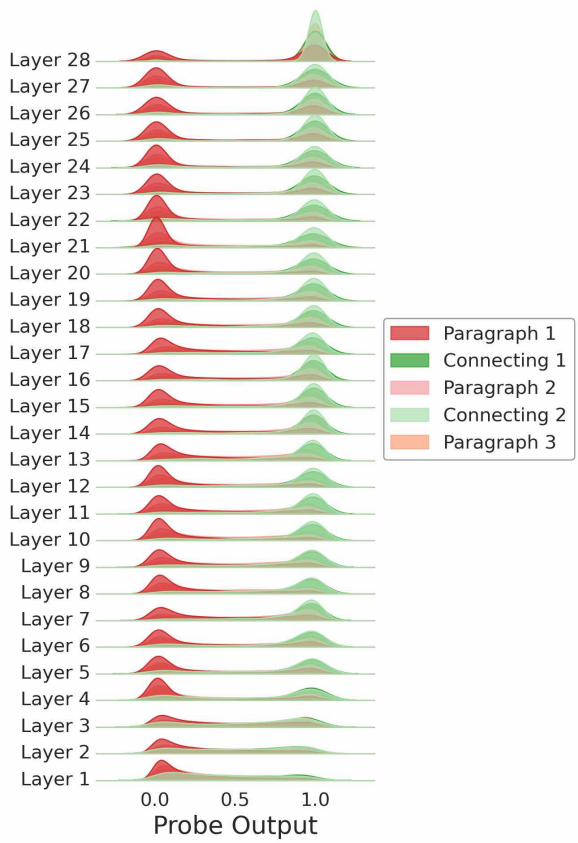

Figure H.36: Layer-wise KDEs for **democracy** probe outputs in `Qwen2.5-7B`

H.3.2 Democracy Probe Results for Best Layers

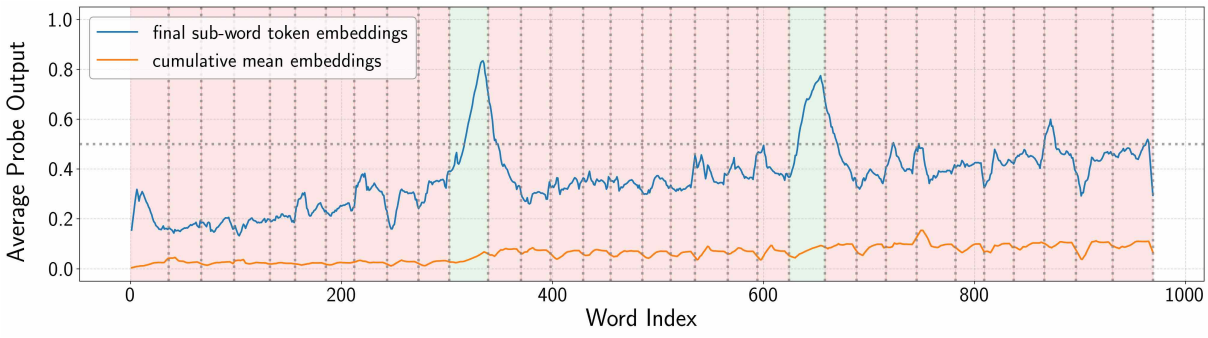

Figure H.37: **Democracy** probe outputs across words using both representative embeddings in `Llama-3-8B`

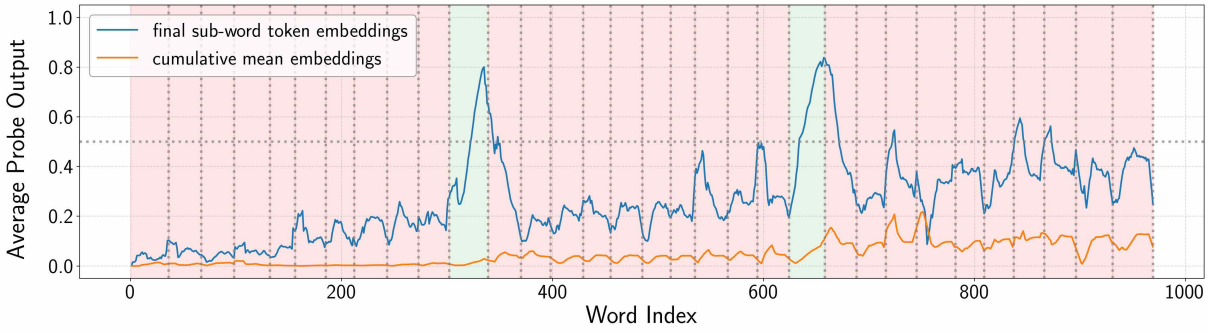

Figure H.38: **Democracy** probe outputs across words using both representative embeddings in `Gemma-2-2B`

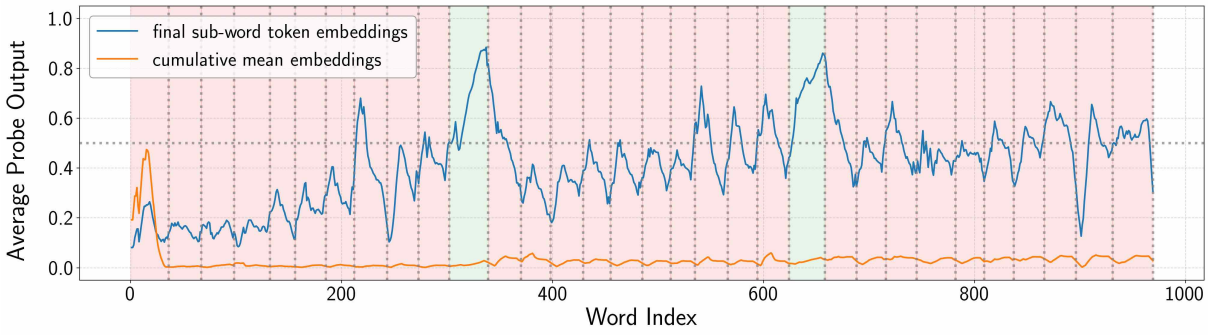

Figure H.39: **Democracy** probe outputs across words using both representative embeddings in `Gemma-2-9B`

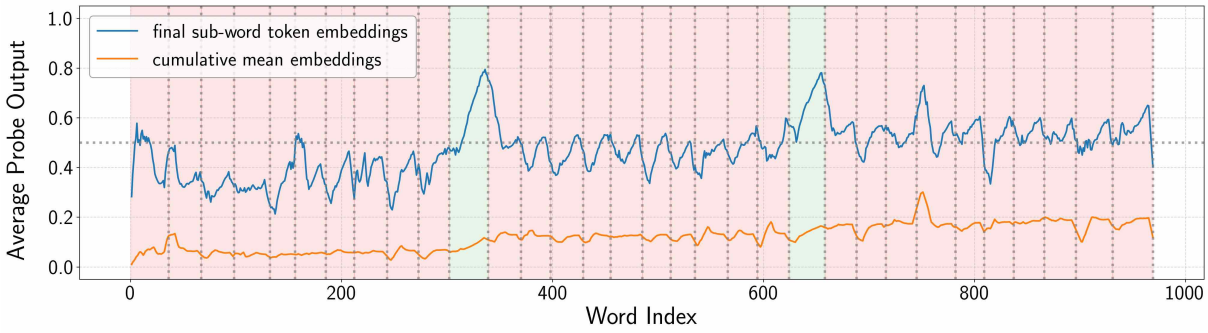

Figure H.40: **Democracy** probe outputs across words using both representative embeddings in `Qwen2.5-0.5B`

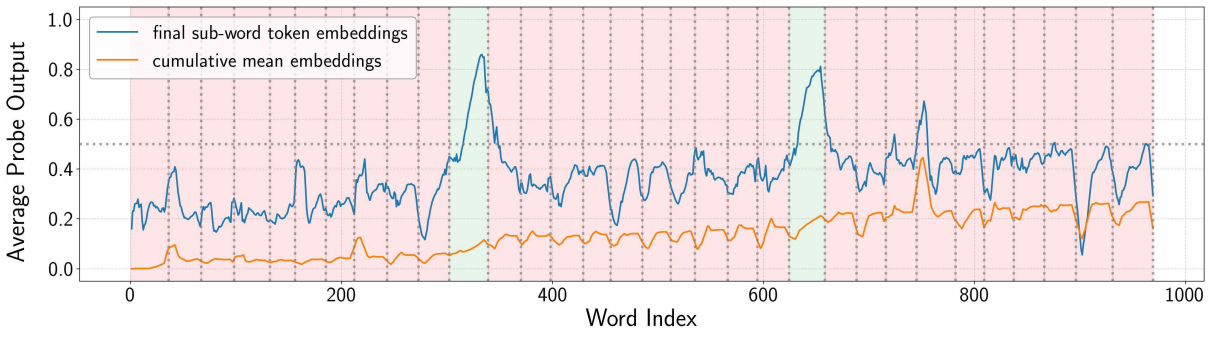

Figure H.41: **Democracy** probe outputs across words using both representative embeddings in `Qwen2.5-1.5B`

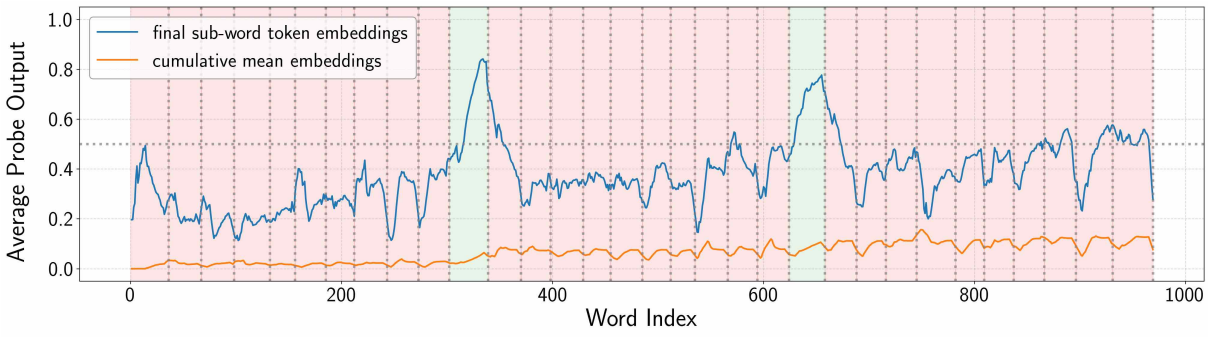

Figure H.42: **Democracy** probe outputs across words using both representative embeddings in `Qwen2.5-3B`

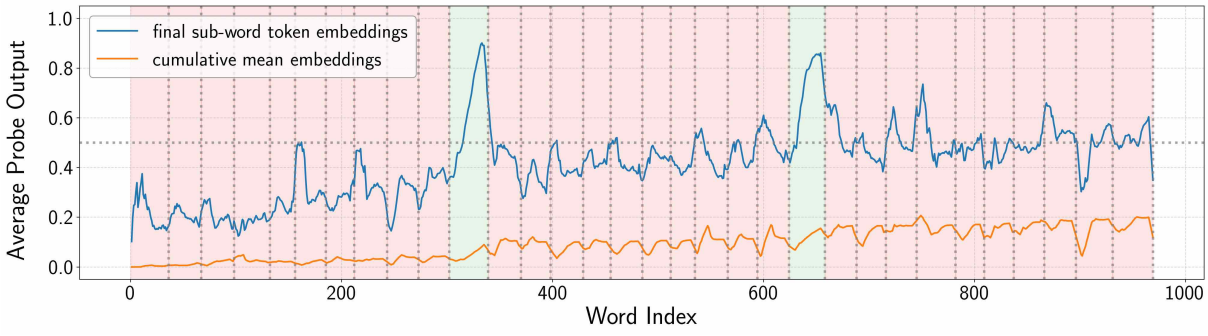

Figure H.43: **Democracy** probe outputs across words using both representative embeddings in `Qwen2.5-7B`

H.4 Tracking Waxing and Waning of Envy

H.4.1 Layer-Wise KDEs for Envy Probe Outputs

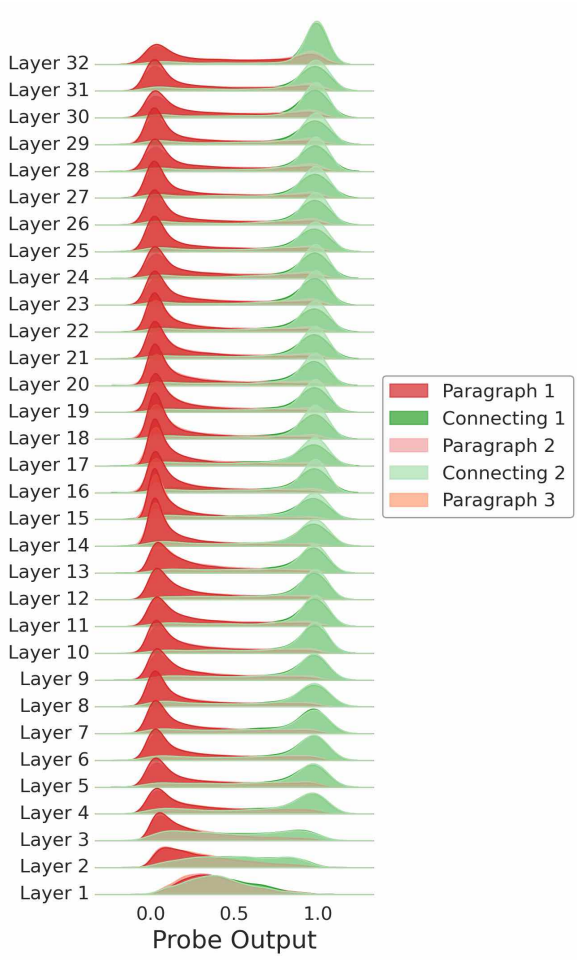

Figure H.44: Layer-wise KDEs for **envy** probe outputs in `Llama-3-8B`

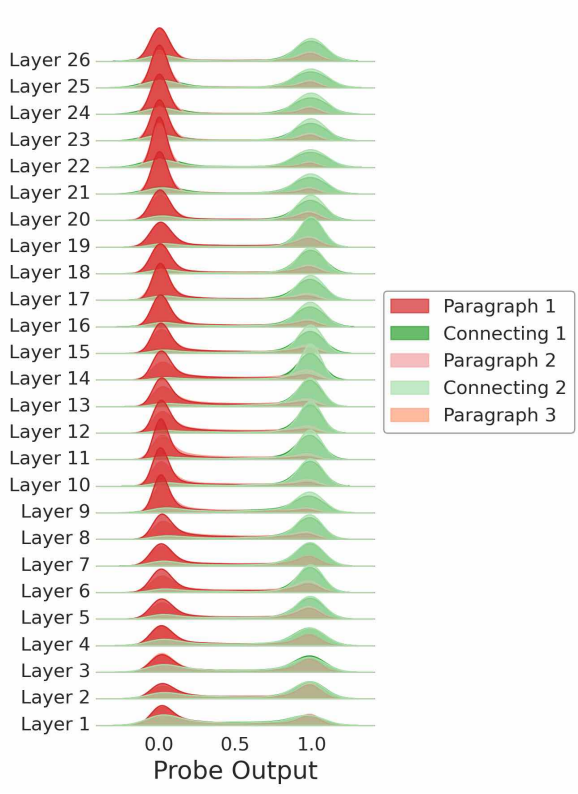

Figure H.45: Layer-wise KDEs for **envy** probe outputs in `Gemma-2-2B`

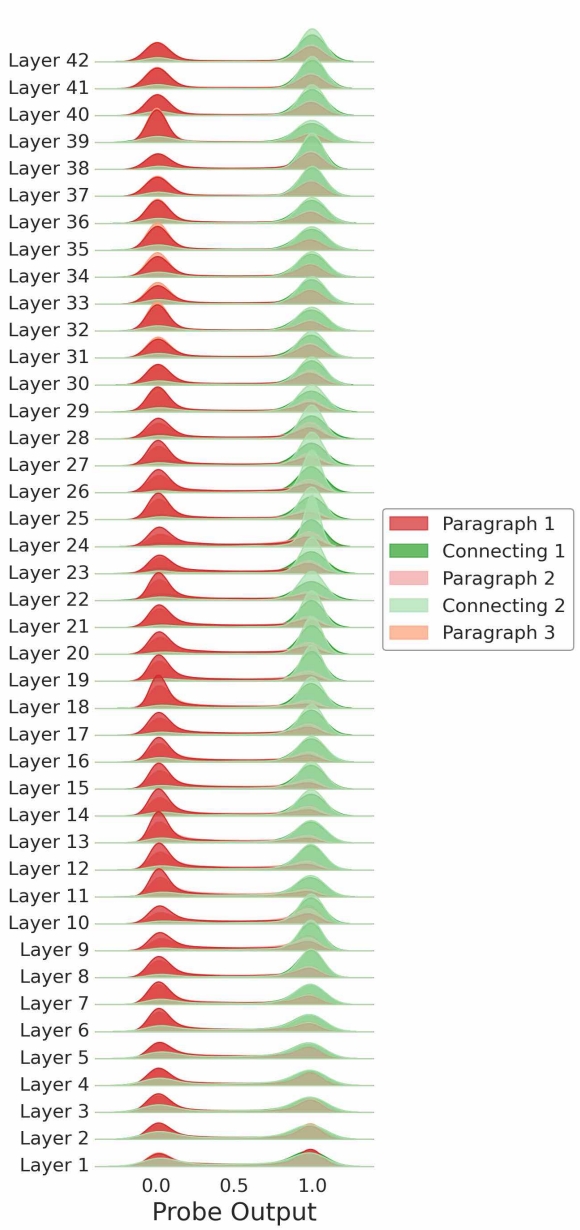

Figure H.46: Layer-wise KDEs for **envy** probe outputs in `Gemma-2-9B`

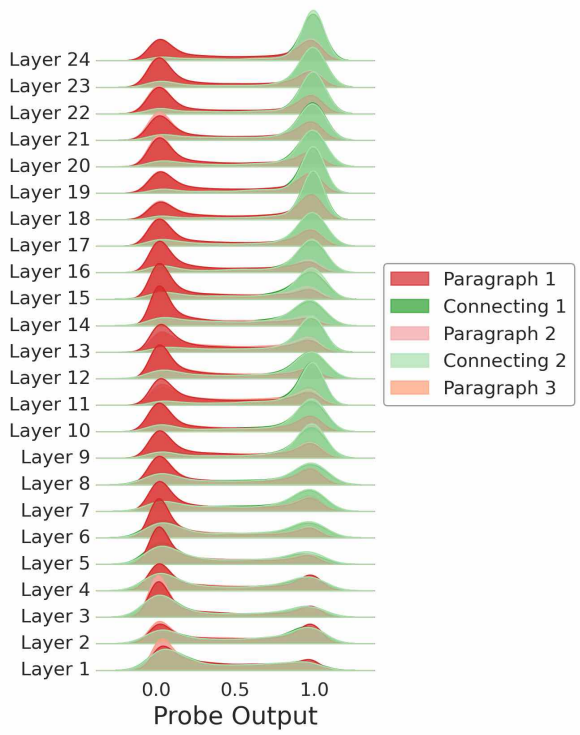

Figure H.47: Layer-wise KDEs for **envy** probe outputs in `Qwen2.5-0.5B`

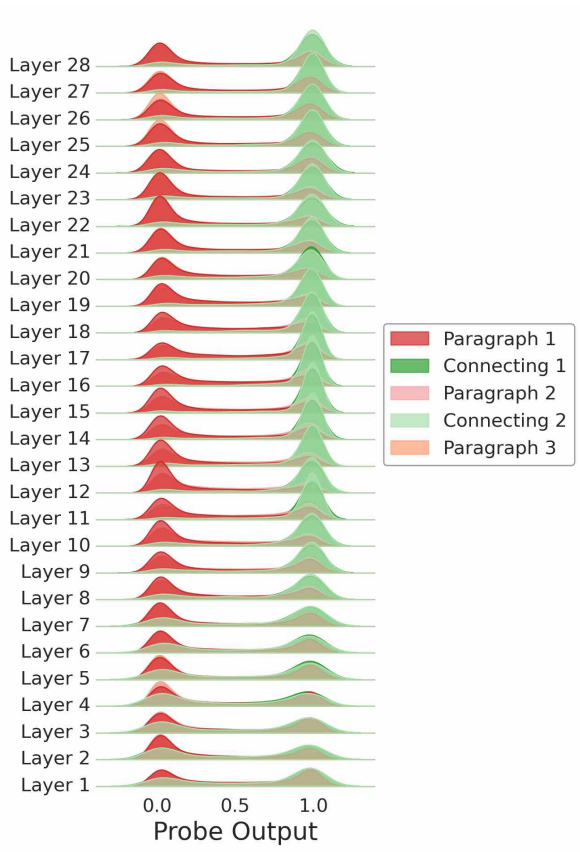

Figure H.48: Layer-wise KDEs for **envy** probe outputs in `Qwen2.5-1.5B`

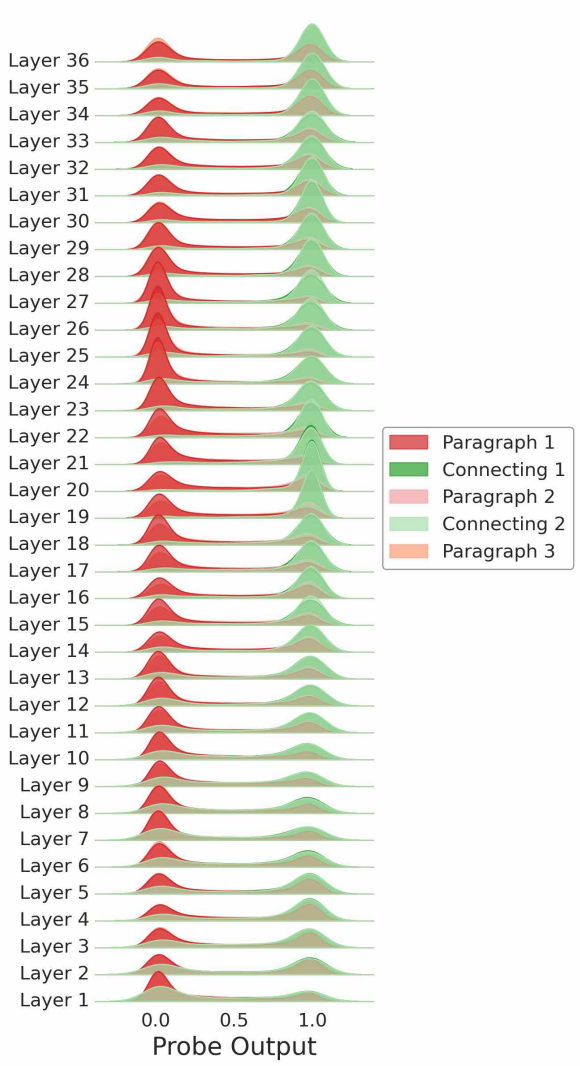

Figure H.49: Layer-wise KDEs for **envy** probe outputs in `Qwen2.5-3B`

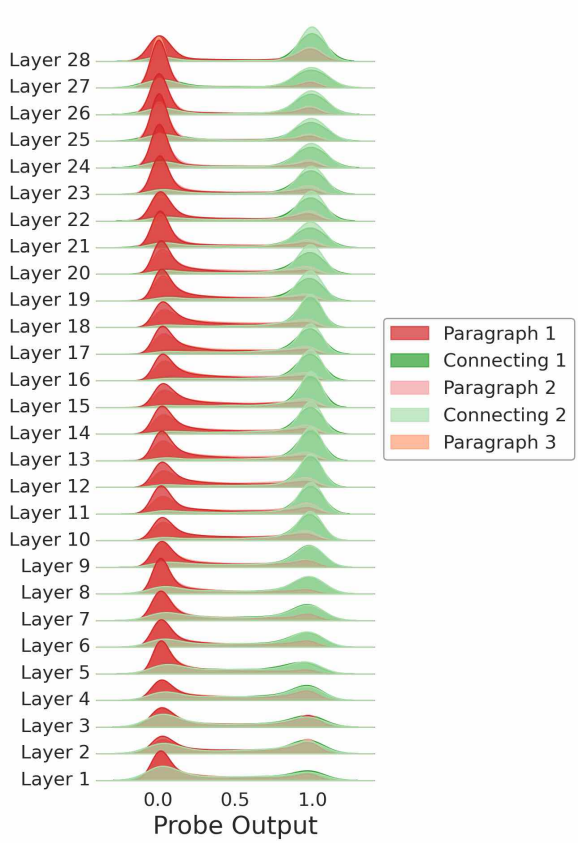

Figure H.50: Layer-wise KDEs for **envy** probe outputs in `Qwen2.5-7B`

H.4.2 Envy Probe Results for Best Layers

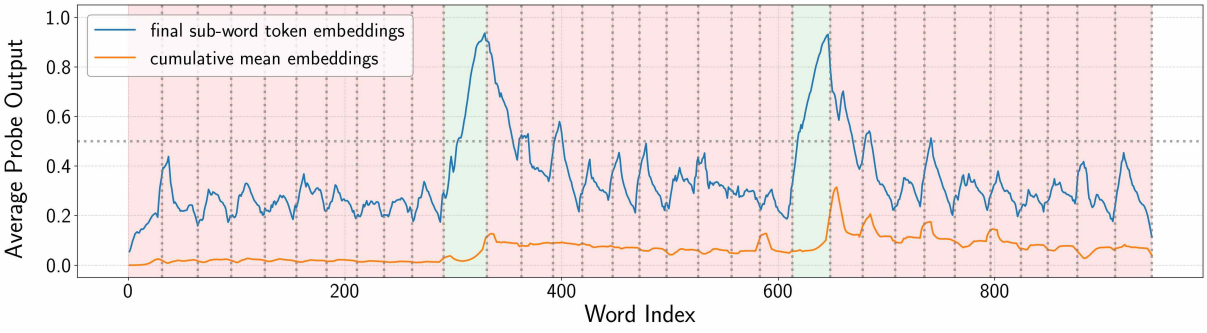

Figure H.51: **Envy** probe outputs across words using both representative embeddings in `Llama-3-8B`

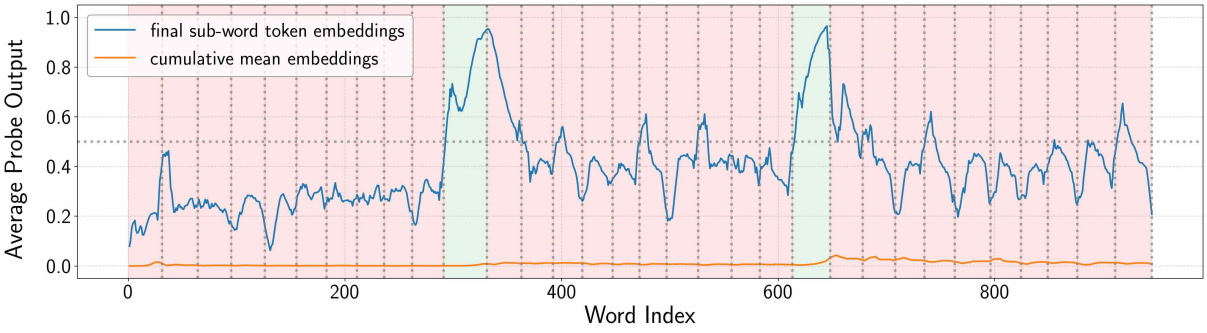

Figure H.52: **Envy** probe outputs across words using both representative embeddings in `Gemma-2-2B`

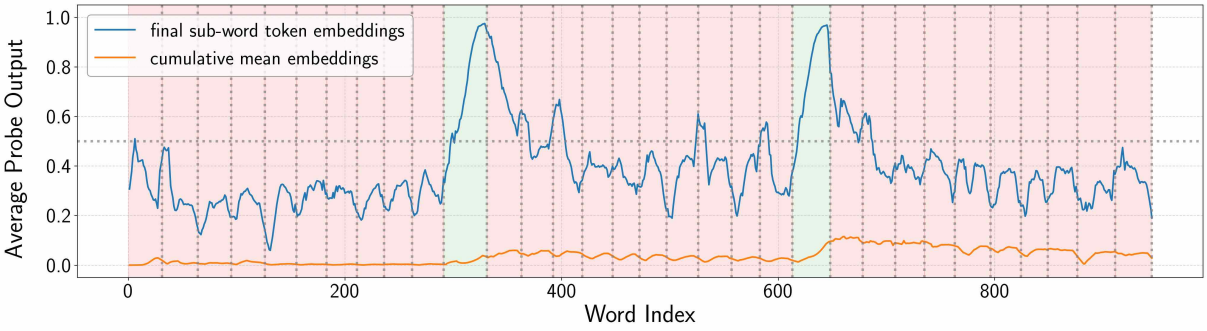

Figure H.53: **Envy** probe outputs across words using both representative embeddings in `Gemma-2-9B`

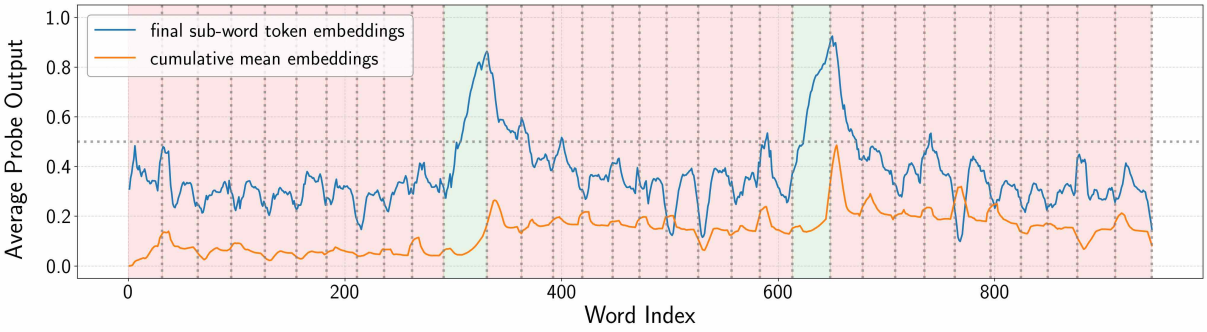

Figure H.54: **Envy** probe outputs across words using both representative embeddings in `Qwen2.5-0.5B`

Figure H.55: **Envy** probe outputs across words using both representative embeddings in `Qwen2.5-1.5B`

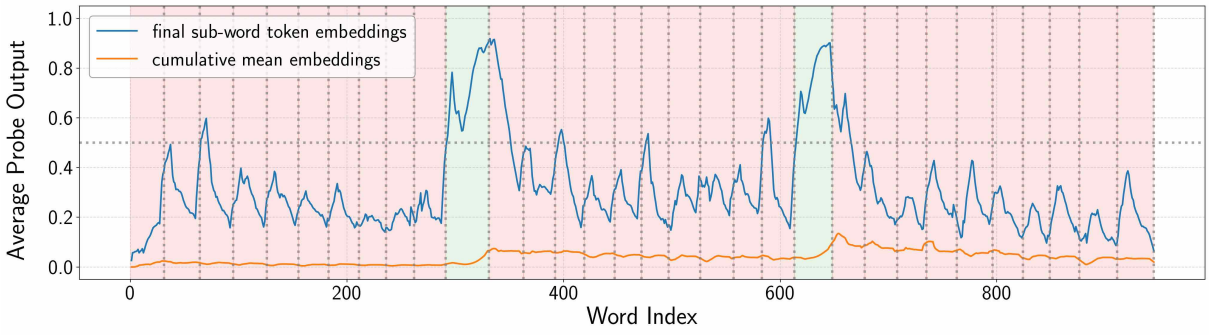

Figure H.56: **Envy** probe outputs across words using both representative embeddings in `Qwen2.5-3B`

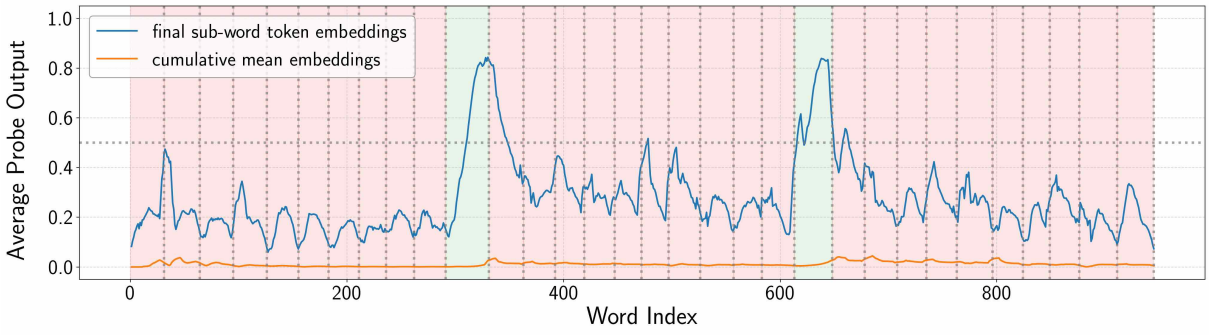

Figure H.57: **Envy** probe outputs across words using both representative embeddings in `Qwen2.5-7B`

I Dataset Overview

We release all the datasets created and used in this paper to be used by others for concept exploration in LLMs. Table I.1 on the next page describes the contents of each file.

File name	General file description	File contents
templates.txt	contains example templates used to create concept datasets	contains one template per line
ambition.csv	contains examples where **ambition** is either present or absent	file has 2 columns: • input_text: contains examples • label: assigned label for each example
investigation.csv	contains examples where **investigation** is either present or absent	file has 2 columns: • input_text: contains examples • label: assigned label for each example
democracy.csv	contains examples where **democracy** is either present or absent	file has 2 columns: • input_text: contains examples • label: assigned label for each example
envy.csv	contains examples where **envy** is either present or absent	file has 2 columns: • input_text: contains examples • label: assigned label for each example
amb_strength.csv	contains 32-sentence stories where **ambition** is present in only two distant sentences per story	file has 2 columns: • input_text: contains stories • label: list of assigned labels for each sentence in each story
inv_strength.csv	contains 32-sentence stories where **investigation** is present in only two distant sentences per story	file has 2 columns: • input_text: contains stories • label: list of assigned labels for each sentence in each story
dem_strength.csv	contains 32-sentence stories where **democracy** is present in only two distant sentences per story	file has 2 columns: • input_text: contains stories • label: list of assigned labels for each sentence in each story
env_strength.csv	contains 32-sentence stories where **envy** is present in only two distant sentences per story	file has 2 columns: • input_text: contains stories • label: list of assigned labels for each sentence in each story

Table I.1: Description of released files for the datasets