Trilemma of Truth in Large Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

The public often attributes human characteristics to large language models (LLMs) and claims that they "know" certain things. LLMs have an internal probabilistic knowledge that represents information retained during training. This study analyzes two common methods for probing the veracity of LLMs and identifies several flawed underlying assumptions. To address these flawed assumptions, we introduce `sAwMIL` (short for Sparse Aware Multiple-Instance Learning). `sAwMIL` uses multiple-instance learning and conformal prediction, while leveraging internal activations of LLMs to classify statements as true, false, or neither. We evaluate `sAwMIL` across 16 open-source LLMs, including both default and chat-based variants, as well as on three new curated datasets. We show that (1) the veracity signal is often concentrated in the third quarter of an LLM's depth; (2) truth and falsehood signals are not always symmetric; and (3) LLMs encode a third type of signal that is distinct from both true and false. These findings provide a reliable method for verifying what LLMs "know" and how certain they are of their probabilistic internal knowledge.

1 Introduction

Can we trust the content that large language models (LLMs) generate? Recent literature suggests that LLMs indeed have internal probabilistic knowledge [1, 2, 3, 4, 5]. However, our understanding of how LLMs use their internal knowledge (if at all) remains fragmented. We know that LLMs are indifferent to the veracity of their outputs [6], and often hallucinate [7]. More than that, it is often difficult for human users to recognize a hallucination because LLMs produce fluent and persuasive texts. For example, Church [8] shows that students trust factually incorrect answers from GPT due to their authoritative and confident tones; and Williams et al. [9] demonstrate that users rate disinformation generated by LLMs as equally or even more credible than human-generated content. Thus, we need a method that can assess the veracity of the internal probabilistic knowledge to improve interactions with LLMs. Thus, We need a way to assess the truthfulness of internal probabilistic knowledge to make LLM interactions more reliable.

Prompt-based evaluations (see Fig. 1A) rely on the idea that we can simply ask LLM about its knowledge. Abbasi Yadkori et al. [10] introduces an information-theoretic prompt-based evaluation, while Xu et al. [11] propose a training framework to produce prompts with self-reflective rationales, and Farquhar et al. [12] introduce uncertainty estimators to detect inconsistent text-generations. However, prompt-based evaluations are sensitive to the input's phrasing [13] and content [14].

A more direct approach is to examine *how* LLMs represent text internally (see Fig. 1B). Consider a large language model, $\mathcal{M}$, with vocabulary $\mathcal{V}$. The LLM maps input text $\boldsymbol{x}$ to a probability distribution over subsequent tokens, denoted $P_{\mathcal{M}}$:

$$\mathcal{M}(\boldsymbol{x}) = P_{\mathcal{M}}(\tau \mid \boldsymbol{x}), \text{ where } \tau \in \mathcal{V} \tag{1}$$

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

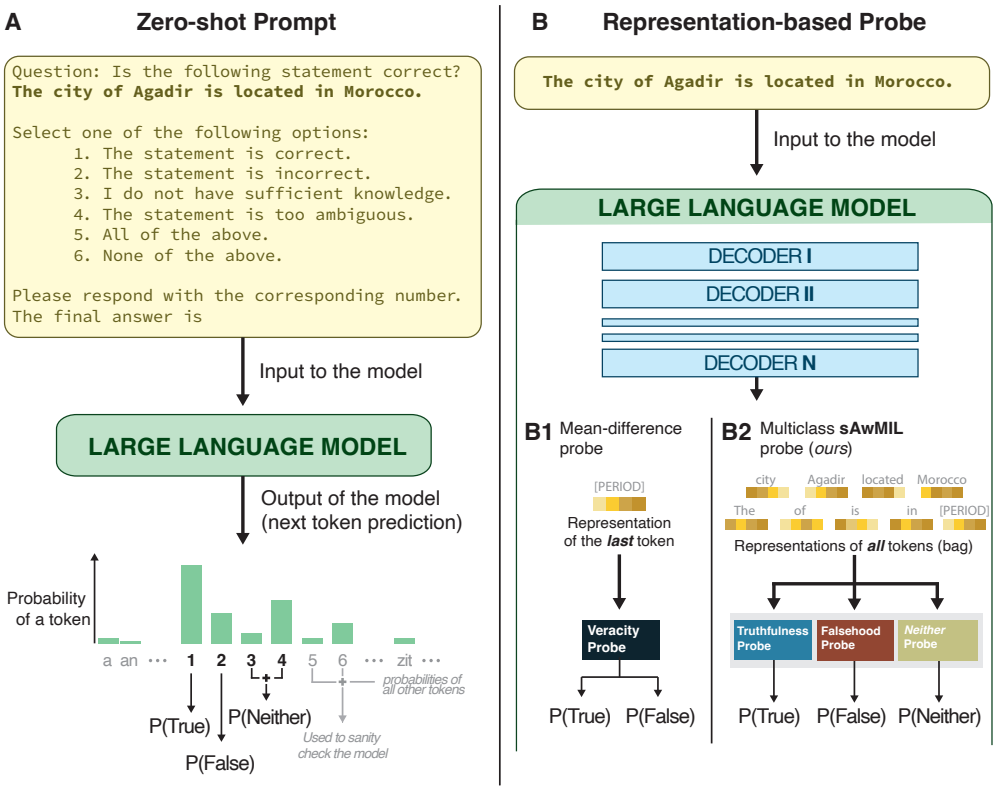

Figure 1: Overview of methods for probing veracity in LLMs. (**A**) In zero-shot prompting, a target statement is inserted into a structured prompt instructing the LLM to select an answer from a specific set of tokens. The LLM's prediction is based on the probabilities of these tokens. This method treats the model as a black box and examines its ⟨input, output⟩ pairs. (**B**) In representation-based probing, the analysis is done on the internal representations generated by intermediate decoders. (**B1**) The mean-difference probe [15] is a common method for determining the veracity of a statement based on the representation of the last token. This approach outputs probabilities for *true* or *false* statements, but cannot account for statements that lack a definitive truth value. (**B2**) Our probe, multiclass sparse aware MIL (`sAwMIL`), looks at the representation of every token in a statement and provides probabilities for three classes: *true*, *false*, and *neither*. Multiclass `sAwMIL` can account for cases when the LLM does not have any knowledge about the statement.

For any token $\tau \in \mathcal{V}$, the distribution $P_{\mathcal{M}}(\tau \mid \boldsymbol{x})$ denotes the probability that τ is the continuation of the sequence $\boldsymbol{x}$. To compute the conditional distribution $P_{\mathcal{M}}$, an LLM transforms $\boldsymbol{x}$ into *intermediate neural activations* denoted $h_i(\boldsymbol{x}) \in \mathbb{R}^{L \times d}$. Here $h_i(\boldsymbol{x})$ denotes the neural activations of $\mathcal{M}$ after the ith decoder, d stands for the hidden dimensionality of the decoder, and L stands for the length of the sequence $\boldsymbol{x}$. We can probe these intermediate activations to identify *veracity signals* – isolating activation patterns that identify truthful statements.[1] For example, Azaria and Mitchell [16] train a neural network to classify statements as *true* or *false* based on these internal representations. Similarly, Marks and Tegmark [15] use a mean-difference classifier to linearly separate the *true* or *false* statements (see Fig. 1B1). Further examples include the unsupervised method introduced by Bürger et al. [17] and a semi-supervised method based on the contrastive pairs of statements [18]. Collectively, these works rely on the idea that given a data set $\langle \boldsymbol{x}, y \rangle \in \mathcal{D}$ with some statements $\boldsymbol{x}$ and veracity labels $y \in Z$, we can train a probe g_i that maps neural activations $h_i(\boldsymbol{x})$ into the distribution $G_{\mathcal{M}}$ over $\mathcal{M}$'s veracity labels:

$$g_i(h_i(\boldsymbol{x})) = G_{\mathcal{M}}(z \mid \boldsymbol{x}), \text{ where } z \in \{\text{true}, \text{false}\} \tag{2}$$

[1]We use the terms 'pattern' and 'signal' interchangeably. Similarly, 'neurons' and 'features' are used to refer to individual components of a signal. In this context, a signal or pattern denotes a set of features that operate collectively.

However, we observe that existing probing methods often rely on flawed assumptions, which limit the reliability of their findings (for overview refer to Supplementary Tab. 4). We argue for a three-valued logic approach (as in Fig. 1B2) as the more appropriate method for modeling veracity in LLMs. Our method `sAwMIL` (short for Sparse Aware Multiple-Instance Learning) combines Multiple Instance Learning (MIL) [19] and Conformal Predictions (CP) [20] to allow for a flexible probe that can handle '*neither*' statements and quantify uncertainty.

In summary, our contributions include the following.

1. We identify and discuss five flawed assumptions in the current veracity-probing literature.
2. We propose a novel multiclass linear probing method `sAwMIL` based on Multiple Instance Learning (MIL) [19] and Conformal Prediction [20, 21].
3. We present three new data sets containing statements labeled *true*, *false*, and *neither*[2] to enable more rigorous evaluations of veracity probes.

2 Background and Flawed Assumptions When Probing Veracity in LLMs

An LLM, $\mathcal{M}$, has **internal probabilistic knowledge** $\mathrm{K}_{\mathcal{M}}$, which it acquires during training.[3] To determine the veracity of a statement ϕ, the model $\mathcal{M}$ should be able to distinguish between three scenarios:

1. ϕ is **True** if there is sufficient support for ϕ given $\mathrm{K}_{\mathcal{M}}$:

$$\mathrm{P}(\phi \mid \mathrm{K}_{\mathcal{M}}) \geq \zeta, \text{ where } \zeta \in (0,\, 1] \text{ is a threshold.}$$

2. ϕ is **False** if there is sufficient support for $\neg\phi$ given $\mathrm{K}_{\mathcal{M}}$:

$$\mathrm{P}(\neg\phi \mid \mathrm{K}_{\mathcal{M}}) \geq \zeta, \text{ where } \zeta \in (0,\, 1] \text{ is a threshold.}$$

3. ϕ is **Neither** if there is not sufficient support for ϕ and $\neg\phi$ given $\mathrm{K}_{\mathcal{M}}$:

$$\Big[\mathrm{P}(\phi \mid \mathrm{K}_{\mathcal{M}}) < \zeta\Big] \text{ and } \Big[\mathrm{P}(\neg\phi \mid \mathrm{K}_{\mathcal{M}}) < \zeta\Big], \text{ where } \zeta \in (0,\, 1] \text{ is a threshold.}$$

If $\mathcal{M}$ has a mechanism to determine the veracity of a statement ϕ, then $\mathcal{M}$ should encode the signal associated with the veracity in its intermediate activations:

1. **Truthfulness**: $\mathcal{M}$ generates an activation pattern that encodes support for ϕ in $\mathrm{K}_{\mathcal{M}}$, reflecting the model's internal support for the statement ϕ.
2. **Falsehood**: $\mathcal{M}$ produces an activation pattern that reflects a lack of sufficient support for ϕ, instead indicating that the internal knowledge $\mathrm{K}_{\mathcal{M}}$ provides stronger support for $\neg\phi$ (e.g., signaling a contradiction or misalignment with known facts).
3. **Neither**: $\mathcal{M}$ should encode the lack of support for ϕ and $\neg\phi$, indicating that the veracity of ϕ is currently undefined. That is, ϕ is neither true nor false.

2.1 Flawed Assumptions When Probing Veracity in LLMs

To train and evaluate a veracity probe g_i , a labeled data set $\mathcal{D}$ is assembled. This data set consists of pairs of neural activations and ground-truth labels, denoted as $\langle h_i(\boldsymbol{x}), y\rangle$, where h_i is the activations after the ith decoder and labels y specify the veracity label Z. In most cases, $Z \in \{\text{true, false}\}$. The probe g_i is trained on the train split $\mathcal{D}_{train} \subseteq \mathcal{D}$ and evaluated on the test split $\mathcal{D}_{test} \subseteq \mathcal{D}$. The intersection between $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ is empty.

We focus exclusively on linear probes, where the parameters of g_i define a linear direction $\vec{\nu}_i$ for the veracity signal after the ith decoder of $\mathcal{M}$:

$$g_i(\boldsymbol{x}) = \boldsymbol{x}\,\boldsymbol{\theta}^T + b \text{ , where } \quad \boldsymbol{\theta} \in \mathbb{R}^{1\times d},\, b \in \mathbb{R} \text{ are parameters learned on } \mathcal{D}_{train} \text{ and } \boldsymbol{x} \in \mathbb{R}^{1\times d}. \tag{3}$$

[2]Throughout the paper, we use the terms *neither*, *neither*-valued, *neither*-type, and *neither-true-nor-false* interchangeably to refer to statements that are neither true nor false. When used as a class label, we italicize *neither* to distinguish it from the regular use of the word. We similarly italicize words such as *true* and *false* when referring to class labels.

[3]Supplementary Tables 2 and 3, respectively, list the notations and abbreviations used in this paper.

Next, we provide a detailed overview of the flawed assumptions made in the existing literature. Refer to the Supplementary Tab. 4 for a condensed overview of flawed assumptions.

Flawed Assumption I: Truth and falsehood are bidirectional. To determine the veracity of a statement ϕ, an LLM $\mathcal{M}$ must develop a mechanism to detect ϕ's truth or falsehood.[4] This mechanism must rely on $\mathcal{M}$'s neural activations to find support for ϕ by using $\mathcal{M}$'s internal probabilistic knowledge $\mathrm{K}_{\mathcal{M}}$. Existing veracity probes [15, 17, 18, 22] implicitly assume that truth and falsehood are encoded bidirectionally. That is,

$$P\left(\phi \mid \mathrm{K}_{\mathcal{M}}\right) = 1 - P\left(\neg\phi \mid \mathrm{K}_{\mathcal{M}}\right) \tag{4}$$

This formulation implies (1) a closed-world assumption, where any statement ϕ not confirmed as *true* is considered *false*, and (2) each decoder symmetrically encodes a signal corresponding to falsehood and truthfulness. However, there is little support to justify either scenario. Similarly, Bürger et al. [17] and Marks and Tegmark [15] suggest that veracity exists along more than two directions.

Valid Assumption I (Truthfulness and falsehood have distinct directions)**.** *The representation of truth and falsehood requires more than one direction. This is,* $P\left(\phi \mid \mathrm{K}_{\mathcal{M}}\right) \neq 1 - P\left(\neg\phi \mid \mathrm{K}_{\mathcal{M}}\right)$.

Flawed Assumption II: LLMs capture and retain everything we know. To train a probe g_i, we use $\mathcal{D}_{train}$ that consists of pairs of factual statements and ground-truth labels $\langle \boldsymbol{x}_i, y_i \rangle$, where (usually) $y \in \{\text{true, false}\}$. The labels y that *we* assign to the statements in $\mathcal{D}$ are based on *our* knowledge (i.e., what we know to be true). Thus, the veracity labels in $\mathcal{D}$ are distributed according to $G_{\mathcal{D}}$.

Our goal, however, is to train a veracity probe g_i that classifies *what the **LLM** deems to be true, false, or 'neither'*. So, the probe g_i should map the statements to the space of the LLM's internal probabilistic knowledge $\mathrm{K}_{\mathcal{M}}$. However, $\mathcal{M}$ may follow a different distribution $G_{\mathcal{M}}$ for the veracity labels. That is, $G_{\mathcal{M}}$ may not be equivalent to $G_{\mathcal{D}}$. For example, we know that "The city of Bissau is in Congo" has a ground-truth label $y =$ false, because we can check maps or official sources. On the other hand, we do not know how $\mathcal{M}$ labels it.

Even though the majority of recent studies use open-source models the precise composition of their training data remains mostly unknown [23, 24]. Even with access to the data, we do not have straightforward methods to verify what has made it into the internal probabilistic knowledge $\mathrm{K}_{\mathcal{M}}$. Thus, the ground-truth label distribution $G_{\mathcal{D}}$ is not necessarily equivalent to the in-model label distribution $G_{\mathcal{M}}$. Recent probing methods [15, 18, 22, 25] cannot account for the mismatch between the label distributions. Instead, these probes introduce a systemic bias, where g_i captures a signal that reflects *our* labeling choices rather than the model's true internal representations.

Valid Assumption II (LLMs do *not* capture and retain everything we know)**.** *The distribution of ground-truth labels $G_{\mathcal{D}}$ may not be equivalent to the model's label distribution $G_{\mathcal{M}}$.*

Flawed Assumption III: All veracity probes provide calibrated probabilities. Veracity probes are generally designed to predict discrete labels. That is, they are classification tasks where the probe assigns one of two labels to a given statement: $g_i : h_i(\boldsymbol{x}) \rightarrow \{\text{true, false}\}$. However, as Herrmann and Levinstein [26] point out, veracity probes should provide not only discrete labels, but also values that can be interpreted as degrees of belief (or some other alternative that quantifies confidence).

Valid Assumption III (The probabilities generated by veracity probes are not inherently calibrated)**.** *The output of veracity probes g_i may not be calibrated and require additional post-processing to be interpreted as meaningful estimates of confidence.*

Flawed Assumption IV: Every statement is either true or false. There are cases where the LLM lacks definitive evidence to determine if a statement is true or false. Suppose we have a veracity probe g_i, which returns a probability of 0.5 for a given statement ϕ to be true. The question is how we should interpret this probability of 0.5. Probes like the mean-difference [15] cannot account for these scenarios. For instance, in Supplementary Sec. H, we show an example where the probe assigns high scores to *pre-actualized* and *neither* statements. To address the issue, we train probes that can account for *neither* cases, so that g_i can reflect the insufficient evidence in $\mathrm{K}_{\mathcal{M}}$. For instance, studies with human participants have shown that including options such as "other" or "I do not know" can help with data quality [27].

[4] In this example, we assume that the veracity label is $Z \in \{\text{true, false}\}$.

Valid Assumption IV (Some statements are neither true nor false)**.** *A probe g_i should distinguish between the cases where the model $\mathcal{M}$ lacks sufficient support to assess the truthfulness of the statement ϕ, and the cases where ϕ lacks a veracity value.*

Flawed Assumption V: We know where the signal for veracity is stored. The majority of veracity probes are trained on the representation of the last token [15, 25]. For example, if the statement is "Boston is in the US.", they assume the period alone carries all of the veracity signal. Such methods assume that any factual signal appearing n tokens before the end of the statement will be faithfully preserved until the last token. A more reasonable approach is to probe at the exact position where the statement is actualized—e.g., immediately after "in the" in the above example—rather than relying on the LLM to *move* that signal all the way to the end of the statement.

Valid Assumption V (Position of the veracity token is not known a priori)**.** *Probes should include a flexible mechanism for identifying the optimal token positions from which to extract veracity signals, instead of relying on fixed positions such as the final token in the statement.*

A probe that directly addresses these flawed assumptions would better reflect the internal knowledge of the LLM and provide a clearer understanding of (1) the factual information encoded in $\mathcal{M}$, (2) how $\mathcal{M}$ classifies statements as *true*, *false*, or *neither*, and (3) calibrated measures of $\mathcal{M}$'s confidence in its own probabilistic knowledge.

3 Method

To address the flawed assumptions, we propose a multiclass probe, called `sAwMIL` (short for *sparse aware multiple-instance learning*). It classifies statements into three classes: *true*, *false*, and *neither*. `sAwMIL` uses multiple-instance learning (MIL) [28] and conformal prediction (CP) [20].

3.1 Sparse Aware Multiple-Instance Learning

Algorithms such as logistic regression, support vector machines, and mean-difference classifiers belong to the single-instance learning (SIL) family, where each instance in the data set is individually labeled. In contrast, multiple-instance learning (MIL) is a type of weakly supervised learning that operates on a set of labeled *bags* [28]. A bag, $\boldsymbol{B}$, is a set of related instances (e.g., patches extracted from the same image or embeddings of individual words in a sentence). Each bag has an associated binary label,[5] but the labels for individual instances within the bag remain unknown. A positive label ($y = 1$) indicates that at least one instance in the bag $\boldsymbol{B}$ belongs to the positive class. Thus, an MIL algorithm must identify the most influential instances contributing to the bag's label. These algorithms must consider the overall structure of the bag and simultaneously suppress irrelevant instances. Bunescu and Mooney [19] introduced sparse balanced MIL (`sbMIL`), an adaptation of linear support vector machines (SVM). It is designed for cases where bags are sparse and only a few instances within a bag are important. `sbMIL` has two training stages. In the first stage, it uses the MIL-modified SVM [19], referred to as sparse MIL (`sMIL`, see Fig. 4 in [19]). During this stage, the objective is to identify the most important instances within positive bags, pushing all other instances and negative bags toward the opposite side of the separating hyperplane. Once the initial model is trained, it computes the distribution of scores assigned to each instance in all positive bags. Then, it computes the η-quantile. η is the *balancing hyperparameter*. Instances scoring above the η threshold are marked as positive. In the second stage, it switches to the single-instance SVM. It works with individual samples, disregarding their original grouping into bags, and assigns them the labels determined during the first stage. We provide the pseudocode for `sAwMIL` in the Supplementary Alg. 1.

3.1.1 Workflow

One-vs-all `sAwMIL`. We modify `sbMIL` since we have an additional piece of information. We know which tokens come from the actualized part of the statement (e.g., "Latvia") and which ones come from the pre-actualized part of the statement (e.g., "The city of Riga is in").

After we apply the η-quantile threshold, we add another round of filtering (see Supplementary Alg. 1, Step 6). Each bag has a set of instances $\boldsymbol{x}_i$, a binary bag label y, and intra-bag labels $\boldsymbol{m}_i$, where

[5] For simplicity, we assume a binary label.

Table 1: Composition of data sets used in this work. Number of *true*, *false*, and *neither*-valued statements per data set. **A** stands for the number of affirmative statements, and **N** stands for the number of negated statements. The last column displays example statements with ground truth labels.

Data Set	True	False	Neither	Examples
City Locations	A: 1392 N: 1376	A: 1358 N: 1374	A: 876 N: 876	(True) The city of Mâcon is located in France. (False) The city of Dharān is located in Ecuador. (Neither) The city of Staakess is located in Marbate.
Word Definitions	A: 1234 N: 1235	A: 1277 N: 1254	A: 1747 N: 1753	(True) Corsage is a synonym of a nosegay. (False) Towner is not a type of a resident. (Neither) Kharter is not a synonym of a greging
Medical Indications	A: 1423 N: 1347	A: 1329 N: 1424	A: 478 N: 522	(True) PR-104 is indicated for the treatment of tumors. (False) Zolpidem is indicated for the treatment of angina. (Neither) Alostat is indicated for the treatment of candigemia.

$\boldsymbol{m}_i \in \{0,1\}^{L_i}$. ($L_i$ is the number of items/tokens in the bag.) These intra-bag labels specify the instances where we expect to find a signal. Given a statement, $\boldsymbol{x} \leftarrow [\boldsymbol{x}^p, \boldsymbol{x}^a]$, all the tokens in the pre-actualized part $\boldsymbol{x}^p$ have an intra-label of 0 (since the factual statement has not yet been actualized), and all the tokens in the actualized part $\boldsymbol{x}^a$ have a label of 1. To label a sample, this sample should have a score above η-quantile, and it should be part of an actualized part $\boldsymbol{x}^a$.

We use `sAwMIL` to train three one-vs-all `sAwMIL` probes that isolate distinct veracity signals:

- `is-true` probe: separates tokens that carry a true signal from all others.
- `is-false` probe: separates tokens that carry a false signal from all others.
- `is-neither` probe: separates tokens that carry neither (not true or false) signal from all others.

Multiclass `sAwMIL` Ideally, we want a multiclass probe that assigns probabilities to a statement being *true*, *false*, or *neither*. Thus, we assemble the one-vs-all `sAwMIL` probes into a multiclass probe via *softmax regression*, which takes the outputs of the one-vs-all probes and transforms them into multiclass probabilities. Formally

$$p_k = \frac{exp(z_k)}{\sum_j exp(z_j)}, \tag{5}$$

where $z_k = g_i^k(\boldsymbol{x}) \cdot \alpha_k + \beta_k$ and $k \in \{\texttt{is-true}, \texttt{is-false}, \texttt{is-neither}\}$.

3.2 Conformal Predictions

Raw outputs from many models, such as Support Vector Machines (SVMs)—specifically, the distance to hyperplane score—are not meaningful as confidence measures. Wrapping SVM scores in a sigmoid function to force them into $[0, 1]$ does *not* create calibrated probabilities as well. They can underestimate their true confidence unless they are explicitly calibrated. Thus, we introduce conformal learning into our probe.

Conformal learning is a framework [20, 21] that enables us to transform raw scores into prediction sets with *guaranteed coverage*. Hence, it provides a method to account for uncertainty. Conformal prediction methods identify intervals within which the probes' predictions are correct with a probability of $1 - \alpha$. For a detailed description of the nonconformity scores [29], see Sec. G in the Supplementary Material.

4 Experiments

This section outlines our experimental setup, including the evaluation procedure, the data sets used, and the selection of large language models.

4.1 Data

We introduce three new data sets consisting of factually *true*, factually *false*, and *neither* statements. The *neither* statements are the ones whose truthfulness value cannot be determined at the present moment (due to the lack of information). While several benchmark data sets for veracity and factuality evaluation exist, prior work has shown that some of these may be partially included in the pretraining or fine-tuning stages of LLMs [30]. In contrast, our goal was to minimize the risk of data contamination while also maintaining higher control over data provenance and quality. Hence, our data sets involve statements related to specific themes (see Tab. 1 for examples):

- **City Locations** data set contains statements about cities and their corresponding countries extracted from the `GeoNames` geographical database.
- **Medical Indications** data set consists of statements about the medications and their corresponding indications from the `DrugBank 5.1` pharmaceutical knowledge base [31]. Medications include the drug and substance names, while indications specify the symptoms or a disease/disorder.
- **Word Definitions** data set is based on the `WordsAPI` dictionary. Hence, the statement involves words and their synonyms or relations.

Every data set consists of negated statements like "The city of Riga **is not** located in Estonia."[6] and affirmative ones like "Menadione **is** indicated for the treatment of coughs."[7] We provide a detailed description of these data sets in the Supplementary Sec. C.

***Neither* statements**. If a statement ϕ is absent from the LLM's internal probabilistic knowledge $\mathrm{K}_{\mathcal{M}}$, then ϕ is *neither* true nor false. It is difficult to determine what statements are absent from $\mathrm{K}_{\mathcal{M}}$ because we generally do not have access to the training data sets used to train the LLMs. However, we can create *neither* statements with *synthetic entities*—i.e., entities that do not exist in the real world or fictional works. Since these objects are specifically generated for our experiments, it is highly unlikely that an LLM has learned anything about them during training. Thus, we can use them as *substitutes* for content that LLMs could not have learned—i.e., from the point of view of an LLM, these should be considered neither *true* nor *false*. For a detailed description of the generation [32] of *neither*-valued statements, see Sec. C.1 in the Supplementary Material.

4.2 Language Models

In our experiments, we use 16 open-source LLMs (ranging from 3 to 14 billion parameters) across 4 families: `Gemma/Gemma-2`, `Llama-3` (`v3.1` and `v3.2`), `Mistral-v0.3`, and `Qwen-2.5`. These models run on consumer-grade hardware and are publicly available through HuggingFace [33]. We provide an overview of these models in Sec. D of the Supplementary Material.

4.3 Evaluation

In this work, we focus on the classification performance to evaluate how well probes can separate 3 classes of statements: *true*, *false*, and *neither*. To do so, we use Matthew's Correlation Coefficient (MCC) to summarize the statistical accuracy of probes (on the test sets); refer to Eq. 15 in the Supplementary Sec. I for the definition of the multiclass MCC. Note, $\mathrm{MCC} = 1$ indicates that a classifier predicted every instance correctly. $\mathrm{MCC} = 0$ implies that the predictions are random. $\mathrm{MCC} = -1$ indicates that the predictions are inversely correlated with the ground-truth labels.

Zero-shot prompting, one-vs-all `sAwMIL`, and multiclass `sAwMIL` can abstain from making predictions. If a probe abstains too often, it suggests poor performance. For these cases, we use Weighted-MCC (W-MCC), where the *acceptance rate* serves as the weight (see Eq. 6).

$$\text{W-MCC} = \mathrm{MCC} \times \left(1 - \frac{\text{\# abstained}}{\text{\# total predictions}}\right) \tag{6}$$

In Supplementary Sec. I, we provide additional results demonstrating `sAwMIL`'s ability to generalize across datasets, and how the identified veracity directions $\vec{\nu}_i$ can be used for targeted interventions on the output token distribution. For brevity, we do not cover these in the manuscript.

[6] This statement is a factually true and negated statement.

[7] This statement is a factually false and affirmative statement.

5 Results

We compare three probing methods: (1) zero-shot prompting, (2) mean-difference probe with conformal prediction intervals (`MD+CP`), and (3) our multiclass `sAwMIL` probe. Overviews of zero-shot prompting and `MD+CP` are provided in Sections F and H.1 of the Supplementary Material. Fig. 2 reports the performance metric for zero-shot prompting, `MD+CP`, and multiclass `sAwMIL` probe.

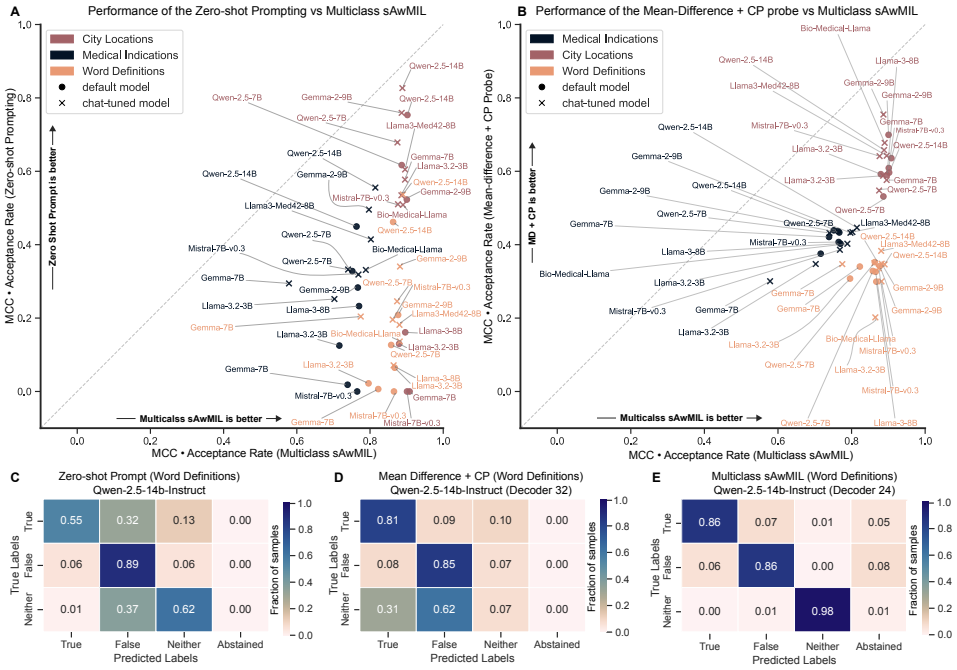

Figure 2: Performances of three probing methods. **Panels A & B:** Each marker shows a probe's performance for a ⟨model, dataset⟩ pair. Default models are shown with circles, while chat models are shown with crosses. The different colors indicate the different data sets. Panel A shows the comparison between the multiclass `sAwMIL` probe on the x-axis and the zero-shot prompting on the y-axis. Panel B shows the comparison between the multiclass `sAwMIL` probe on the x-axis and the mean-difference probe with conformal prediction intervals (`MD+CP`) on the y-axis. For the zero-shot prompting and the multiclass `sAwMIL` probe, we report Weighted Matthew's Correlation Coefficient, and for the `MD+CP` we report the default MCC value. **Panels C–E:** They show confusion matrices for the `Qwen-2.5-14` (chat) model on the *Word Definitions* data set. Overall, multiclass `sAwMIL` probe outperforms zero-shot prompting and `MD+CP`, especially when it comes to the separation of the *neither*-valued statements (see panels C–E).

Zero-shot prompting. Fig. 2A shows that multiclass `sAwMIL`, with its representation-based probes, outperforms zero-shot prompting. We also observe that zero-shot prompting is less accurate for default models. It achieves the best performance on the relatively simple *City Locations* data set, but performs worse on the other two data sets. It also disproportionately predicts *false*: around a third of *true* and a third of *neither* statements are classified as false (see example in Fig. 2C). We see a similar skew in the confusion matrices of other models with zero-shot prompting (see Supplementary Tab. 16).

Mean-difference probe with conformal prediction intervals (`MD+CP`). Fig. 2B shows that multiclass `sAwMIL` outperforms `MD+CP`. Unlike zero-shot prompting, the probes for the default and chat models exhibit a smaller difference in performance for representation-based probes. We further see that `MD+CP` does not perform well on *neither* statements (see Fig. 2D and Supplementary Tab. 17 for more details). It captures either a proxy or a mixture of signals corresponding to some other properties rather than veracity.

Multiclass `sAwMIL` probe. As Figures 2A and 2B show the multiclass `sAwMIL` probe has the best overall performance. First, there are no significant performance differences between the chat and

default models (as is the case with the `MD+CP` probe). Second, except for the `Gemma-7B` (a chat model) on *Medical Indications*, the multiclass `sAwMIL` probes achieve W-MCC values higher than 70%, indicating strong performance. The confusion matrix in Fig. 2E demonstrates that `sAwMIL` has good separation between the *true* and *false* statements, and the probe correctly separates almost all of the *neither* statements. This ability to provide better separation of *true*, *false*, and *neither* statements is observed across all the models (for details, see Supplementary Tab. 18). Also, Sec. H.2 in the Supplementary Material contains results for the single-instance SVM.

5.1 At which layer is the linear representation of veracity concentrated?

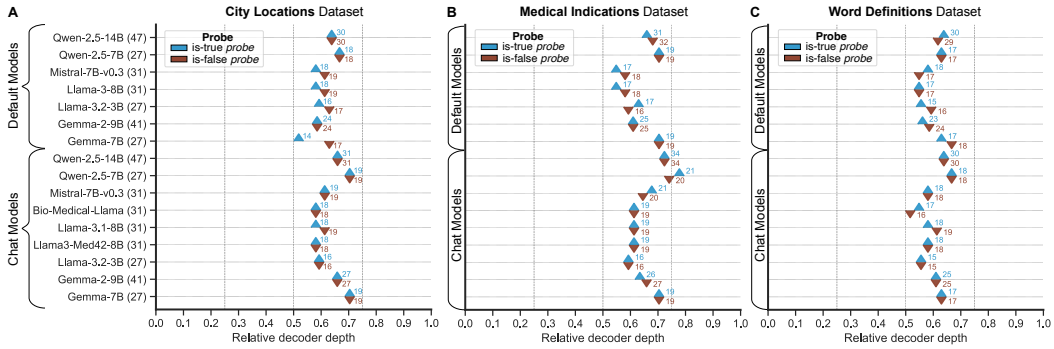

Figure 3: Decoders with median classification performance for the one-vs-all `sAwMIL` probes. The number in the parentheses next to each model name indicates the total number of decoders for that model. For example, the `Qwen-2.5-14B` default model has 47 decoders (i.e., 47 layers). We mark decoders based on their associated W-MCC values. Each triangle marks the decoder at which the cumulative metric reaches 50% of its total across all decoders. Thus, in Panel A, the `is-true` probe for the `Qwen-2.5-14B` default model has a median decoder of 30, meaning that the sum of W-MCC values over decoders 1–30 equals that over decoders 31–47. **Panels A–C**: Results for the three data sets across 16 LLMs. First, the veracity signal usually resides between 0.5 and 0.75 in the relative decoder depth. Second, classification performance indicates a slight mismatch between the signal strengths for *true* and *false* statements.

Fig. 3 depicts the **location of decoders based on median classification performance**. If all decoders encoded the veracity information equally well, we would expect the median-performing decoder to occur around the midpoint of the model's depth (i.e., at relative decoder depth of 0.5). However, as Panels A–C of Fig. 3 show the median performing decoders are located between 0.5 and 0.75 in the relative decoder depth of models. In particular, the veracity probes consistently perform better on deeper decoders for `Qwen` and `Gemma` (chat) models compared to `Llama` and `Mistral` models. Note that this pattern is consistent across all the data sets in Panels A–C of Fig. 3.

When the median-performing decoders of the `is-true` and `is-false` probes coincide, this suggests a unified linear direction for truth and falsehood. That is, both truth and falsehood signals are *equally* present at every given decoder. This alignment is frequently observed across chat models. However, a mismatch often appears in the default models, where truth and falsehood are encoded asymmetrically across decoders or emerge at different depths within the model. Our analysis of the interventions shows a further discrepancy between the signal alignment (see Sec. I.2 in the Supplementary material).

6 Conclusion

In this work, we take a critical look at popular methods for probing the veracity of large language models (LLMs) and identify flawed assumptions underlying them. To address these flaws, we introduce `sAwMIL`, a multiclass linear probe that combines Multiple Instance Learning with Conformal Prediction Intervals. Unlike prior methods, `sAwMIL` models veracity using three classes: *true*, *false*, and *neither*. Across sixteen models and three datasets, `sAwMIL` outperforms existing probes and provides new insights into how veracity signals are localized within LLMs, revealing that they tend to concentrate in the third part of the network.

References

[1] Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17817–17825, 2024.

[2] Haowen Pan, Xiaozhi Wang, Yixin Cao, Zenglin Shi, Xun Yang, Juanzi Li, and Meng Wang. Precise localization of memories: A fine-grained neuron-level knowledge editing technique for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[3] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[4] David Chanin, Anthony Hunter, and Oana-Maria Camburu. Identifying linear relational concepts in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1524–1535, 2024.

[5] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona T Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *CoRR*, 2022.

[6] Michael Townsen Hicks, James Humphries, and Joe Slater. ChatGPT is bullshit. *Ethics and Information Technology*, 26(2):38, 2024.

[7] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55, 2025.

[8] Kenneth Church. Emerging trends: When can users trust GPT, and when should they intervene? *Natural Language Engineering*, 30(2):417–427, 2024.

[9] Angus R Williams, Liam Burke-Moore, Ryan Sze-Yin Chan, Florence E Enock, Federico Nanni, Tvesha Sippy, Yi-Ling Chung, Evelina Gabasova, Kobi Hackenburg, and Jonathan Bright. Large language models can consistently generate high-quality content for election disinformation operations. *PloS one*, 20(3): e0317421, 2025.

[10] Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37:58077–58117, 2024.

[11] Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. Sayself: Teaching LLMs to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5985–5998, 2024.

[12] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[13] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, 2022.

[14] Reid McIlroy-Young, Katrina Brown, Conlan Olson, Linjun Zhang, and Cynthia Dwork. Order-independence without fine tuning. *Advances in Neural Information Processing Systems*, 37:72818–72839, 2024.

[15] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=aajyHYjjsk`.

[16] Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023.

[17] Lennart Bürger, Fred A Hamprecht, and Boaz Nadler. Truth is universal: Robust detection of lies in LLMs. *Advances in Neural Information Processing Systems*, 37:138393–138431, 2024.

[18] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=ETKGuby0hcs`.

[19] Razvan C Bunescu and Raymond J Mooney. Multiple instance learning for sparse positive bags. In *Proceedings of the 24th international conference on Machine learning*, pages 105–112, 2007. doi: 10.1145/1273496.1273510.

[20] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

[21] Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=eNdiU_DbM9`.

[22] Benjamin A Levinstein and Daniel A Herrmann. Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*, pages 1–27, 2024.

[23] Abhinav Atla, Rahul Tada, Victor Sheng, and Naveen Singireddy. Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges*, 26(5):96–103, 2011.

[24] Shivani Gupta and Atul Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019.

[25] Jacqueline Harding. Operationalising representation in natural language processing. *British Journal for the Philosophy of Science*, 2023. doi: 10.1086/728685.

[26] Daniel A Herrmann and Benjamin A Levinstein. Standards for belief representations in LLMs. *Minds and Machines*, 35(1):1–25, 2025. doi: 10.1007/s11023-024-09709-6.

[27] Sara Dolnicar and Bettina Grün. Including Don't know answer options in brand image surveys improves data quality. *International Journal of Market Research*, 56(1):33–50, 2014.

[28] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[29] Ulf Johansson, Henrik Linusson, Tuve Löfström, and Henrik Boström. Model-agnostic nonconformity functions for conformal classification. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2072–2079. IEEE, 2017.

[30] Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models. *CoRR*, 2024.

[31] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

[32] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2025. URL `https://web.stanford.edu/~jurafsky/slp3/`. Online manuscript released January 12, 2025.

[33] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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, 2020.

[34] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, 2019.

[35] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2025.

[36] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, 2018.

[37] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023. doi: 10.48550/arXiv.2310.01405.

[38] Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

[39] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020. ISBN 978-1-108-45514-5.

[40] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

A Notations and Abbreviations

Table 2: Notations used throughout the paper. Symbols are grouped by category: model definitions, inputs and datasets, internal representations, veracity distributions, and intervention-related symbols.

Symbol	Description	Shape / Notes
$\mathcal{M}$	Large language model	
$\mathrm{K}_{\mathcal{M}}$	Internal probabilistic knowledge of the model $\mathcal{M}$	
$\mathcal{V}$	Vocabulary of the model $\mathcal{M}$, consists of tokens	$[\tau_1, \ldots, \tau_{\lvert\mathcal{V}\rvert}] \in \mathcal{V}$
$P_{\mathcal{M}}(\tau \mid \boldsymbol{x})$	Output of $\mathcal{M}$: a conditional probability distribution on tokens	
	Inputs and Datasets	
$\mathcal{D}$	Dataset of statements	$\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}$
$\boldsymbol{x}$	Input token sequence, e.g., "The city of Riga is in Latvia."	$L = \lvert\boldsymbol{x}\rvert$
$\boldsymbol{x}^p$	Pre-actualized part of a statement, e.g., "The city of Riga is in"	
$\boldsymbol{x}^a$	Actualized part of a statement, e.g., "Latvia."	
$\boldsymbol{r}$	Random sequence with length $\lvert\boldsymbol{x}^a\rvert$	$\lvert\boldsymbol{r}\rvert = \lvert\boldsymbol{x}^a\rvert$
y	Veracity label assigned to $\boldsymbol{x}$	$y \in Z$
ϕ	A statement evaluated for veracity	
$\mathcal{T}_S$	Transition matrix for n-gram generation	See Eq. 7
	Internal Representations	
d	Size of the hidden representation (of a decoder)	
$h_i(\boldsymbol{x})$	Activations after the ith decoder	$\mathbb{R}^{L\times d}$
$h_i(\boldsymbol{x})_{[j]}$	Activation of the token at index j after ith decoder	$\mathbb{R}^{1\times d}$ and $j \in \{1 \ldots L\}$
$h_i(\boldsymbol{x})_{[n:m]}$	Activations from nth to mth tokens after ith decoder	$\mathbb{R}^{(m-n)\times d}$
	Veracity, Probes and Distributions	
Z	Set of veracity labels, e.g., {*true*, *false*, *neither*}	
$G_{\mathcal{D}}(z \mid \boldsymbol{x})$	Distribution of veracity labels $z \in Z$ in a dataset $\mathcal{D}$	
$G_{\mathcal{M}}(z \mid \boldsymbol{x})$	Distribution of veracity labels $z \in Z$ in the model $\mathcal{M}$	
g_i	Veracity probe trained on activations of the ith decoder	$g_i : h_i(\boldsymbol{x}) \mapsto G_{\mathcal{M}}$
$\vec{\nu}_i$	Linear direction extracted from the probe g_i	$\mathbb{R}^{1\times d}$
η	Balancing hyperparameter for `sAwMIL`	$\eta \in (0, 1)$
$\boldsymbol{m}$	`sAwMIL`'s intra-bag labels (i.e., labels per-token in each $\boldsymbol{x}$)	$\boldsymbol{m} \in \{0, 1\}^L$, $L = \lvert\boldsymbol{x}\rvert$
	Interventions and Effects (Sec. I.1.2)	
I_i^+	Modified representation of $h_i(\boldsymbol{x}^a)$ after adding $+\vec{\nu}_i$	
I_i^-	Modified representation of $h_i(\boldsymbol{x}^a)$ after subtracting $-\vec{\nu}_i$	
ΔI_i^+	Change in $P_{\mathcal{M}}$ for $\boldsymbol{x}^a$ after I_i^+ intervention	
ΔI_i^-	Change in $P_{\mathcal{M}}$ for $\boldsymbol{x}^a$ after I_i^+ intervention	
$s_j(\boldsymbol{x})$	Per-statement success of the intervention; indicator function	See Eq. 22
$\Delta_{correct}$	Total change in $P_{\mathcal{M}}$ for $\boldsymbol{x}^a$ after both I_i^+ and I_i^- interventions	
Δ_{random}	Total change in $P_{\mathcal{M}}$ for $\boldsymbol{r}$ random tokens after both interventions	
$\mathbb{E}[\Delta_{correct}]$	Average probability difference across all statements	
$\overline{d}_i$	Indicator of the dominant direction for the i-th decoder	$\overline{d}_i \in \{-1, 1\}$

Table 3: Abbreviations and naming conventions used throughout this paper.

Abbreviation	**Full Form**	**Description**
LLM	Large Language Model	
SIL	Single-Instance Learning	Probes on one embedding per example
MIL	Multiple-Instance Learning	
SVM	Support Vector Machine	
DPO	Direct Preference Optimization	LLM finetuning method
RLHF	Reinforcement Learning from Human Feedback	LLM finetuning method
CP	Conformal Prediction Intervals	Uncertainty calibration method
MD+CP	Mean-Difference with Conformal Prediction Intervals	MD probe with abstention via conformal intervals
sAwMIL	Sparse Aware MIL probe	Multiclass probe handling unknowns
MCC	Matthews Correlation Coefficient	Multiclass performance measure (see Eq. 15)
W-MCC	Weighted-MCC	Using Acceptance Rate as a weight (see Eq. 6)

B Assumptions

Tab. 4 provides an overview of the flawed assumptions in the recent probing methods.

Table 4: **Overview of flawed assumptions in recent methods that probe veracity, their impact on reliability, and our corrective strategies.** Probes that do not account for these issues may lead to biased or unreliable findings.

Flawed Assumptions	**Why It Matters**	**Our Solution/Approach**
Truth and falsehood are bidirectional.	There is no conclusive evidence that LLMs treat truth and falsehood as one continuous bidirectional concept. It is more likely that there exist three separate concepts: `is-true`, `is-false`, and `is-neither`; and they have their own distinct mechanisms.	`sAwMIL` is a multiclass probe that treats "*true*," "*false*," and "*neither*" as separate categories.
LLMs capture and retain everything we know.	We do not know what LLMs have been exposed to during training. Consequently, linear probes that assume every fact in a data set is stored within the LLM are prone to systematic errors in their predictions. We must distinguish between what the LLM *actually* retains and what *we* know to be true or false. If a statement $\boldsymbol{x}$ is unknown to the LLM, it is neither true nor false. In such cases, passing $\langle h_i(\boldsymbol{x}), \text{true}\rangle$ or $\langle h_i(\boldsymbol{x}), \text{false}\rangle$ to the probe during training introduces error.	`sAwMIL` is a linear probe that identifies samples with high support before fitting the linear separator.
All veracity probes provide calibrated probabilities.	Probes such as SVM or mean-group difference classifiers often make a prediction based on the sign (w.r.t. the separation hyperplane). We cannot use these scores to evaluate certainty around the predictions. In other words, these probes are rarely calibrated.	`sAwMIL` integrates conformal prediction to quantify uncertainty and produce statistically valid prediction regions.
Every token (or statement) is either true or false.	Not every token or sentence expresses a complete factual claim. We should be able to create probes that refrain from making predictions when there is insufficient support.	Instead of training probes to distinguish between true and false statements, `sAwMIL` is a multiclass classifier that separates statements into "true," "false," and "neither."
We know *a priori* where to look for veracity-related signals.	Most existing probes assume that the last token of a statement has all the information about the veracity.	By using multiple-instance learning, `sAwMIL` is able to select parts of the input that have the most information about veracity.

C Data Sets

We introduce three new data sets: *City Locations*, *Medical Indications*, and *Word Definitions*. Each dataset consists of statements that are factually *true*, factually *false*, or *neither*. These datasets contain both affirmative and negated statements. An example of a false negated statement is "Guaifenesin is **not** indicated for the treatment of coughs", and an example of the true affirmative statement is "Shouter is a type of a communicator."

Data Splits. We split each data set into train, calibration, and test sets using *approximately* 55/20/25 ratios (see Supplementary Tab. 5). We ensure that the objects mentioned in statements are exclusive to the split. For example, if Singapore is mentioned in a statement of the training set, all the statements with Singapore are moved to the training split.

Table 5: Dataset splits. The number of statements per split. In the brackets, we specify the fraction of the total number of statements.

Dataset	Train	Calibration	Test	Total
City Locations	3999 (.55)	1398 (.19)	1855 (.26)	7252 (1.00)
Medical Indications	3849 (.56)	1327 (.19)	1727 (.25)	6903 (1.00)
Definitions	4717 (.55)	1628 (.19)	2155 (.25)	6500 (1.00)

C.1 'Neither' Statements

Since we do not have access to the training data sets of LLMs, we cannot validate whether LLMs retained information about specific facts or entities. That is, we do not know the composition of the internal knowledge $\mathrm{K}_{\mathcal{M}}$ of an LLM. Hence, we cannot be certain about what each LLM can (and cannot) verify. To overcome this issue, we create *neither* statements with *synthetic entities*—i.e., entities that do not exist in the real world or fictional works. The *neither* statements are the ones whose value cannot be determined at present (e.g., due to lack of information).

Generation of 'Neither' Statements

We use synthetic names to generate *neither*-type statements. For example, "The city of *Staakess* is located in *Soldovadago*" mentions a town and a country that do not exist. From the point of view of an LLM, these statements should be considered *neither-true-nor-false*, as LLMs could not have learned anything about these.

To generate the *neither* statements, we use the Markov-Chain technique [32]. Given a set of existing words $[w_1, w_2 \ldots, w_n] \in S$, we break each word w_i into n-grams, For instance, we break "ability" into the following 2-grams: `[start]a ab bi il li it ty y[end]`. We then compute a transition matrix $\mathcal{T}_S$, which provides the probability of transitioning from the n-gram i to n-gram j is given by:

$$\mathcal{T}_S\left(j \mid i\right) = \frac{\text{count}\left(i \to j\right)}{\sum_x \text{count}\left(i \to x\right)} \tag{7}$$

In addition, we use $\mathcal{T}_S$ to sample new synthetic words that follow the n-gram distribution of words in S. In our experiments, we use 3-grams for most entities, except for country names, which we generate with 2-grams. We use the `namemaker`[8] package that implements a Markov-Chain word generator.

C.2 Data Selection and Processing

Next, we provide details on the source and processing steps for each dataset.

[8] github.com/Rickmsd/namemaker

City Locations

The *City Locations* dataset is based on the `GeoNames`[9] database. `GeoNamesCache`[10] is a Python package that interacts with the `GeoNames` API. We use the following criteria to select a ⟨city, country⟩ pair:

1. The population of the city is at least 30,000.
2. The city has an associated country. If a city name is associated with multiple countries, we include the ⟨city, country⟩ pair for each country. We exclude all cities that have "Antarctica" as a location or a country.

Since the resulting set of ⟨city, correct country⟩ pairs is relatively large. We reduce the number of pairs by downsampling. In total, we select 1,400 unique city names: 700 cities with the highest populations, and 700 cities randomly sampled from the rest of the names.

Statement Structure. For each ⟨city, correct country⟩ pair, we create statements of the form:

The city of `[city]` is (not) located in `[country]`.

If a city name already contained a word "city" (e.g., "Guatemala City"), we do not start a sentence with "The city of." We also sample ⟨city, incorrect country⟩ pairs, and generate statements according to the template above.

Synthetic Entities. We use the technique described in Supplementary Sec. C.1 to generate synthetic city and country names. To generate synthetic *city* names, we collect all the city names in our data set (including those that we did not include) and input them to `namemaker` (with n-gram length of 3). We generate 500 synthetic city names. We validate these synthetic names in two stages:

1. We check whether a synthetic name exists in the `GeoNames` database by looking for matches in the *name* and *alternative name* fields. We keep 310 cities after this first stage.
2. We use Google Search to validate that each synthetic city name does not exist via the following prompt: "city [city name]". If the search result returns a city with 1-2 character difference, we remove the synthetic name from the list. We keep 219 cities after this second stage.

For the synthetic country names, we collect all the country names and input them to `namemaker` (with n-gram length of 2). We generate 250 synthetic names and validate them using the workflow described in the previous paragraph. We keep 238 country names after the first stage, and 138 after the second stage. The Google Search prompt is: "country [country name]"). With 25% probability, we add a prefix or suffix to the synthetic country name. The list of prefixes and suffixes include "Island," "Republic of," "Kingdom," "West," "East," "North," "South," and "Land." Finally, we randomly match each synthetic name to the name of a synthetic country.

Medical Indications

The *Medical Indications* dataset is based on the `DrugBank` (version 5.1.12) [31]. We obtain access to the DrugBank on October 4th, 2024, via the academic license (for research purposes only). Our GitHub and Zenodo repositories do not contain the raw data from the DrugBank, but the reader can apply for the academic license.[11] We extract 2 fields from this knowledge base:

1. **Name**, which specifies the official name of the drug or the chemical (e.g., Lepirudin).
2. **Indication**, which is a text field that describes the indication of the drug. If this field consists of multiple sentences, we keep only the first sentence (e.g., "Lepirudin is indicated for anticoagulation in adult patients with acute coronary syndromes (ACS) such as unstable angina and acute myocardial infarction without ST elevation.")

To extract diseases and conditions from the *Indications* field, we use two named entity recognition (NER) models:

[9] geonames.org

[10] pypi.org/project/geonamescache/

[11] Here is the link to the DrugBank's academic license: https://go.drugbank.com/releases/5-1-12

1. `SciSpacy`'s `en_ner_bc5cdr_md` model[12] for the biomedical term annotations
2. `BioBERT`-based NER[13] for disease annotations

We input the "Indication" text to both models. The disease/condition terms are extracted only if both models mark it as a disease or condition. For example, for "Lepirudin is indicated for anticoagulation in adult patients with acute coronary syndromes (ACS)," the `SciSpacy` model marks *coronary syndromes* as a disease, but `BioBERT` does not. Thus, we do not add it to Lepirudin's disease/condition list. Similarly, we remove the abbreviation if the disease list contains the full name *and* its abbreviation, such as [acute coronary syndromes, ACS].

We further validate the drug names via `SciSpacy` model, and keep the name only if it is marked as CHEMICAL. Otherwise, we remove the drug from our dataset. Finally, if the disease list (for a given drug) is empty after the preprocessing, we remove the drug from our data set.

Additionally, we use `wordfreq`[14] package to check whether the name of the drug or the name of the indication appears in widely used corpora (e.g., WikiPedia or Books dataset). In other words, we remove the pair if either the drug name or the indication has a Zipf's frequency of 0 – i.e., the word does not appear in any of the `wordfreq` corporas.

Statement Structure. For each $\langle$drug, correct disease$\rangle$ pair, we create statements of the form:

`[drug]` is (not) indicated for the treatment of `[disease/condition]`.

We also sample the $\langle$drug, incorrect disease$\rangle$ pairs. We ensure that the "incorrect disease" did not share any words with the diseases in the correct list.

Synthetic Entities. To generate synthetic drug names and disease names, we use the approach described in Supplementary Sec. C.1 (with n-gram length of 3). We generate 500 synthetic drug names. We validate these synthetic names in two stages:

1. We pass each generated name through `SciScapy` model and remove the ones marked as `CHEMICAL`. We keep 315 name after this first stage.
2. We use Google Search to validate that each drug name does not exist via the prompt "medicine [drug name]." If the search result returned a drug with 1-2 character difference, we remove it from the dataset. We keep 243 names after this second stage.

We generate 200 disease names and check whether they exist in our list of diseases. We keep 181 names after this first stage. Next, we use Google Search with the prompt "disease [disease/condition name]." We keep 131 disease names after this second stage. Finally, we randomly match synthetic drug names to synthetic disease names to generate *neither*-type statements.

Word Definitions

The *Word Definitions* dataset is based on the sample data from WordsAPI[15] database. Sample data is publicly available and contains 10% of randomly sampled words from the database.[16]

For each word in the sample, we keep the ones that satisfy the following criteria:

1. The word is a noun.
2. The word has at least one definition in the *definition* field.
3. The word has at least one of the following fields: *synonym*, *typeOf*, or *instanceOf*.

Statement Structure. Depending on the specified field (i.e.,*synonym*, *typeOf*, *instanceOf*), we generate three types of statements:

1. "`[word]` is (not) `[instanceOf]`."

[12] allenai.github.io/scispacy/
[13] alvaroalon2/biobert_diseases_ner
[14] pypi.org/project/wordfreq
[15] WordsAPI.com
[16] We do not provide a copy of the sample in our GitHub or Zenodo repositories.

2. "`[word]` is (not) a type of `[typeOf]`."
3. "`[word]` is (not) a synonym of `[synonym]`."

Before inserting a word from *synonym*, *typeOf*, *instanceOf* fields into a corresponding spot, we check which article goes before 'a' or 'an'. When possible, we change words into singular forms. To do so, we use the `inflect` package,[17]

Synthetic Entities. To generate synthetic entities, we use the approach described in Supplementary Sec. C.1 (with n-gram length of 3). We generate four categories of synthetic entities:

1. Words that go at the beginning of each statement: We use all the words we have in the dataset.
2. Types: We use all the words from the *typeOf* field for the Markov-Chain generation.
3. Synonyms: We use all the words from the `synonym` field.
4. Instances: We use words from the *instanceOf* field.

We generate 1,000 synthetic words for each of the four categories. We validate the non-existence of words. We use the `english_words` package[18] to check whether a word exists in "GNU Collaborative International Dictionary of English 0.53," or `web2` word list. Furthermore, we check whether there is a word in the *words* list of the `nltk` package.[19] After this stage, we end up with 3,305 words. Finally, we randomly sample pairs of $\langle$word, property$\rangle$, where the property is a type, instance, or synonym.

[17] pypi.org/project/inflect/
[18] pypi.org/project/english-words/
[19] pypi.org/project/nltk/

D Selection of Large Language Models

In this section, we provide an overview of the large language models used in our experiments. Supplementary Tab. 6 provides a list of all the 16 models.

We use default models—i.e., the ones that were pre-trained on general tasks. We also use chat models that have been fine-tuned on instruction- and chat-like interactions. Every default model in our selection has a corresponding chat-based model. We also add two extra chat-tuned `Llama` models that are specifically fine-tuned on biomedical data. Further, we do not use full official model names but use short names along with a version, such as "chat" or "default". For example, `Llama-3.2` (chat) refers to the `Llama-3.2-3b-Instruct` model.

Table 6: **List of LLMs used in our experiments.** We provide the official names of the models in the HuggingFace repository. Further, we provide the *type* of the model: default stands for the pre-trained models, and 'chat' stands for the chat- or instruction-tuned version of the models. Finally, we provide the number of decoders, the number of parameters, the release date, and the source of the model. These models are publicly available through HuggingFace [33].

Official Model Name	Type	# Decoders	# Parameters	Release Date	Source
Gemma-7b	Default	28	8.54 B	Feb 21, 2024	Google
Gemma-2-9b	Default	26	9.24 B	Jun 27, 2024	Google
Llama-3-8b	Default	32	8.03 B	Jul 23, 2024	Meta
Llama-3.2-3b	Default	28	3.21 B	Sep 25, 2024	Meta
Mistral-7B-v0.3	Default	32	7.25 B	May 22, 2024	Mistral AI
Qwen2.5-7B	Default	28	7.62 B	Sep 19, 2024	Alibaba Cloud
Qwen2.5-14B	Default	38	14.80 B	Sep 19. 2024	Alibaba Cloud
Gemma-7b-it	Chat	28	8.54 B	Feb 21, 2024	Google
Gemma-2-9b-it	Chat	26	9.24 B	Jul 27, 2024	Google
Llama-3.2-3b-Instruct	Chat	28	3.21 B	Sep 25, 2024	Meta
Llama-3.1-8b-Instruct	Chat	32	8.03 B	Jul 23, 2024	Meta
Llama3-Med42-8B	Chat	32	8.03 B	Aug 12, 2024	M42 Health
Bio-Medical-Llama-3-8B	Chat	32	8.03 B	Aug 11, 2024	Contact Doctor
Mistral-7B-Instruct-v0.3	Chat	32	7.25 B	May 22, 2024	Mistral AI
Qwen 2.5-7B-Instruct	Chat	28	7.62 B	Aug 18, 2024	Alibaba Cloud
Qwen 2.5-14B-Instruct	Chat	38	14.80 B	Aug 18, 2024	Alibaba Cloud

E Criteria for Validating Veracity Probe

Table 7: **Validity criteria for representation-based probes.** If satisfied, these criteria serve as validation that g_i indeed captures signals associated with veracity Z. Here, we provide a formal definition of each criterion, along with the implications of satisfying the criterion. Finally, we provide the list of similar criteria and concepts used in the literature.

Criteria	Definition	If Satisfied	Similar Concepts
Correlation	A probe g_i trained on $\langle h_i(\boldsymbol{x}), y\rangle \in \mathcal{D}_{\text{train}}$ should perform well (i.e., have high predictive accuracy) on $\mathcal{D}_{\text{test}}$, assuming the same input and label distributions.	$\mathcal{M}$ encodes information correlated with veracity (see Fig. 2).	Information [25], Accuracy [26]
Generalization	A probe g_i trained on $\langle h_i(\boldsymbol{x}), y\rangle \in \mathcal{D}_{\text{train}}$ should have high predictive accuracy on data from different domains.	$\mathcal{M}$ has a universal activation pattern correlated with veracity (see Fig. 13).	Generalization as defined by Bürger et al. [17], Uniformity [26]
Selectivity	A probe g_i trained on $\langle h_i(\boldsymbol{x}), y\rangle \in \mathcal{D}_{\text{train}}$ should not assign *true* or *false* labels to samples where truthfulness is absent or undefined.	$\mathcal{M}$ has a distinct mechanism that correlates exclusively with veracity (see Fig. 2).	Misrepresentation as defined by Harding [25], Control Task [34]
Manipulation	Modifying $h_i(\boldsymbol{x})$ along $\vec{\nu}_i$ should systematically alter $P_{\mathcal{M}}(\tau \mid \boldsymbol{x})$ for tokens τ related to the veracity property Z.	$\mathcal{M}$ has a *linear* mechanism to track veracity and uses it to compute the output $P_{\mathcal{M}}(\tau \mid \boldsymbol{x})$ (see Fig. 14).	Use [25], Addition [35], Intervention [3]
Locality	Modifying $h_i(\boldsymbol{x})$ along $\vec{\nu}_i$ should not significantly alter $P_{\mathcal{M}}(r \mid \boldsymbol{x})$ for random tokens r that are unrelated to Z.	$\mathcal{M}$ maintains a separate mechanism that tracks veracity, without being confused with other concepts.	Misrepresentation [25], Leakage [36]

Researchers have proposed criteria to measure the validity of veracity probes [25, 26, 37]. We aggregate these into five major categories and provide an overview in Supplementary Tab. 7. We propose to evaluate a probe g_i along the following criteria:

(i) **Correlation**. The probe, trained to predict a veracity property $\{\textit{true}, \textit{false}\} \in Z$, should achieve high predictive accuracy on unseen samples that possess this property, i.e., on samples from $\mathcal{D}_{test}$. When the criterion is satisfied, the i-th decoder embeds the information about Z to some degree. We cannot rule out the fact that it captures proxies associated with Z.

(ii) **Generalization** extends *Correlation* by requiring that the probe generalizes beyond the data set it was trained on. The probe should have high predictive accuracy on samples that have veracity Z, but have different phrasing or come from different domains. For example, if a probe is trained to identify neural activation patterns associated with veracity on statements related to ecology, this probe should have similar predictive accuracy on statements related to biology.

(iii) **Selectivity** The probe g_i should avoid classifying statements that are not (or cannot be) *true* or *false*. Hence, the probe g_i should abstain from making predictions on the *neither*-valued statements—i.e., statements that the LLM could not have learned from its training data or that inherently lack any truthfulness or falsehood. Poor selectivity indicates that the probe might capture spurious correlations with unrelated properties.

(iv) **Manipulation**. We should be able to use the identified direction $\vec{\nu}_i$ to update $h_i(\boldsymbol{x})$ and have a predictable change in the distribution of the output tokens $P_{\mathcal{M}}(\tau \mid \boldsymbol{x})$. Since we focus on *linear* probes, we expect that moving $+c$ units along $\vec{\nu}_i$ should have an opposite effect on $P_{\mathcal{M}}$ compared to moving $-c$ units.

(v) **Locality** When asking a question such as "Is X true? Answer yes or no," the manipulation should primarily influence the generation process related to the "yes" or "no" responses. It should minimally affect unrelated tokens. For example, if a manipulation does not increase the likelihood of the LLM generating "no", but increases the likelihood of generating tokens such as "elephant", then the manipulation degrades the LLM's abilities.

Evaluating a probe according to these criteria allows us to determine how well g_i captures the signals associated with veracity Z and how manipulations (a.k.a. interventions) affect LLM's output $P_{\mathcal{M}}$. Part of our future work includes adding a new criterion on whether the probe can assess if an LLM can "reason" logically. For example, if $\mathcal{M}$ classifies a statement ϕ_1 as true and another statement ϕ_2 as true, then will $\mathcal{M}$ also classify $\phi_1 \wedge \phi_2$ as true?

Finally, we demonstrate the evaluation results for the **Correlation** and **Selectivity** in the Results section (see Sec. 5) of the manuscript; we further provide evaluation results for the **Generalization**, **Manipulation** and **Locality** in the Supplementary Sec. I.

F Zero-Shot Prompting: Instructions, Veracity Labeling, and Abstention

Instructions in Zero-shot Prompting. In zero-shot prompting, each LLM receives an instruction along with a statement x as input. These instructions outline the task and describe the format of the output. We use zero-shot prompts to evaluate how well LLMs can assess the veracity of a given statement and provide a correct reply. Our zero-prompts are designed as a multiple-choice question, where an LLM is asked to pick a corresponding number.

In our (zero-shot prompting) experiments, we use three different templates. All default models share the same template as displayed in Supplementary Fig. 4A–B. We do not use this template for chat-models, since they support "turn-based" conversations. Thus, for most chat models, we use templates that model user-assistant interaction. We use the template in Supplementary Fig. 4D–E for the chat models that support context prompts. We use the template in Supplementary Fig. 4F for `Gemma` models since they do not support context prompts.

Additionally, we use three different phrasings of the instructions:

1. **Original instructions** are displayed in Supplementary Fig. 4A and D. In the main paper, we report results based on these instructions.
2. **True-False instructions**, where we change the phrasing of the original instructions by replacing the words "correct" and "incorrect" to "true" and"false". These are displayed in Supplementary Fig. 4B and E.
3. **Shuffled instructions**, where we change the original instructions by shuffling choice identifiers. These are displayed in Supplementary Fig. 4C and F.

Supplementary Figures 5–7 depict the Weighted Matthew's Correlation Coefficient (W-MCC) performance of zero-shot prompting across 3 datasets and 16 LLMs for each template type and instruction type. In some cases, minor changes to the instructions can have a significant impact on performance. Overall, the `Qwen` and `Gemma-2` models provide the best performance compared to other LLMs. However, `Qwen` models seem to be more sensitive to the phrasing of the prompt.

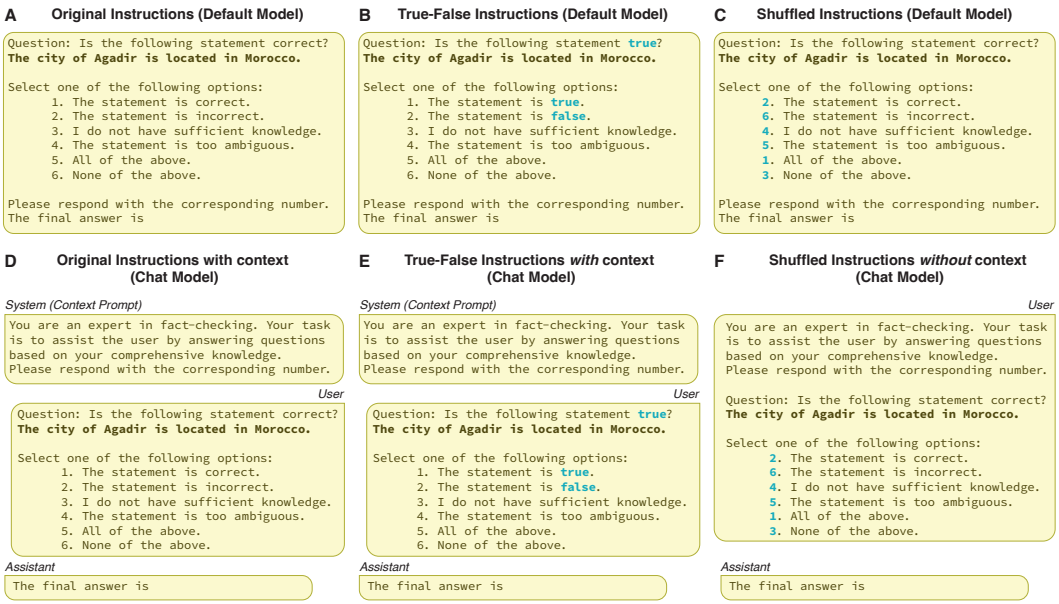

Figure 4: **Zero-shot prompt templates.** We use these templates in our experiments. **Panel A–B:** The prompts for the default models. **Panel D-F:** Examples of prompts used for the chat models. Note that chat models like `Gemma` do not have a context (or system) prompt; hence, we provide instructions in the first message (see Panel F). In the main manuscript, we report the performance over the *original instructions* – i.e., instructions in Panels A and D. For the chat LLMs without the context prompt, we apply the two-message template in Panel F, but use the *original instructions*. (Side note: The statement used in these examples is factually correct.)

From Token Probabilities to Veracity Labels in Zero-shot Prompting. Given the original instructions and the statement $\boldsymbol{x}$, an LLM outputs token-level probabilities over its vocabulary $\mathcal{V}$ as

$$P_{\mathcal{M}}(\tau \mid \text{instruction} \wedge \boldsymbol{x}) \text{ with } \sum_{\tau \in \mathcal{V}} P_{\mathcal{M}}(\tau \mid \text{instruction} \wedge \boldsymbol{x}) = 1.$$

We are interested in the probabilities of the tokens that correspond to the multiple choices – i.e., numbers 1–6 in any of the panels in Supplementary Fig. 4. We denote tokens associated with these numbers as: `[1]`, `[2]`, `[3]`, etc. We map these token-level probabilities $P_{\mathcal{M}}$ into the veracity-label probabilities $G_{\mathcal{M}}$ as follows:

$$G_{\mathcal{M}}(\text{true} \mid \boldsymbol{x}) = P_{\mathcal{M}}(\texttt{[1]} \mid \text{instruction} \wedge \boldsymbol{x}) \tag{8}$$

$$G_{\mathcal{M}}(\text{false} \mid \boldsymbol{x}) = P_{\mathcal{M}}(\texttt{[2]} \mid \text{instruction} \wedge \boldsymbol{x}) \tag{9}$$

$$G_{\mathcal{M}}(\text{neither} \mid \boldsymbol{x}) = P_{\mathcal{M}}(\texttt{[3]} \mid \text{instruction} \wedge \boldsymbol{x}) + P_{\mathcal{M}}(\texttt{[4]} \mid \text{instruction} \wedge \boldsymbol{x}) \tag{10}$$

Abstention in Zero-shot Prompting. We include options `[5]` and `[6]` to check the "sanity" of the model $\mathcal{M}$. For example, option #5 in Supplementary Fig. 4A suggests that a statement $\boldsymbol{x}$ is true, false, and ambiguous – all at the same time. If the model assigns most of the probability mass to these tokens, we assume that the model does not follow the instructions. Similarly, if a model assigns most of its probability mass $P_{\mathcal{M}}$ to other tokens in the vocabulary $\{\tau \in \mathcal{V} : \tau \notin \{\texttt{[1]}, \texttt{[2]}, \texttt{[3]}, \texttt{[4]}, \texttt{[5]}, \texttt{[6]}\}\}$, we also assume that the model did not follow the instructions. Hence, if instructions are not followed, we assume that the model *abstains* from making a prediction, see Eq. 11.

$$G_{\mathcal{M}}(\text{abstain} \mid \boldsymbol{x}) = \sum_{\tau} P_{\mathcal{M}}(\tau \mid \text{instruction} \wedge \boldsymbol{x}), \text{ where } \{\tau \in \mathcal{V} : \tau \notin \{\texttt{[1]}, \texttt{[2]}, \texttt{[3]}, \texttt{[4]}\}\} \tag{11}$$

Note that the zero-shot prompting relies only on the token-level probabilities, i.e., $\mathcal{M}$'s output. It does not look at the intermediate hidden representation of $\boldsymbol{x}$.

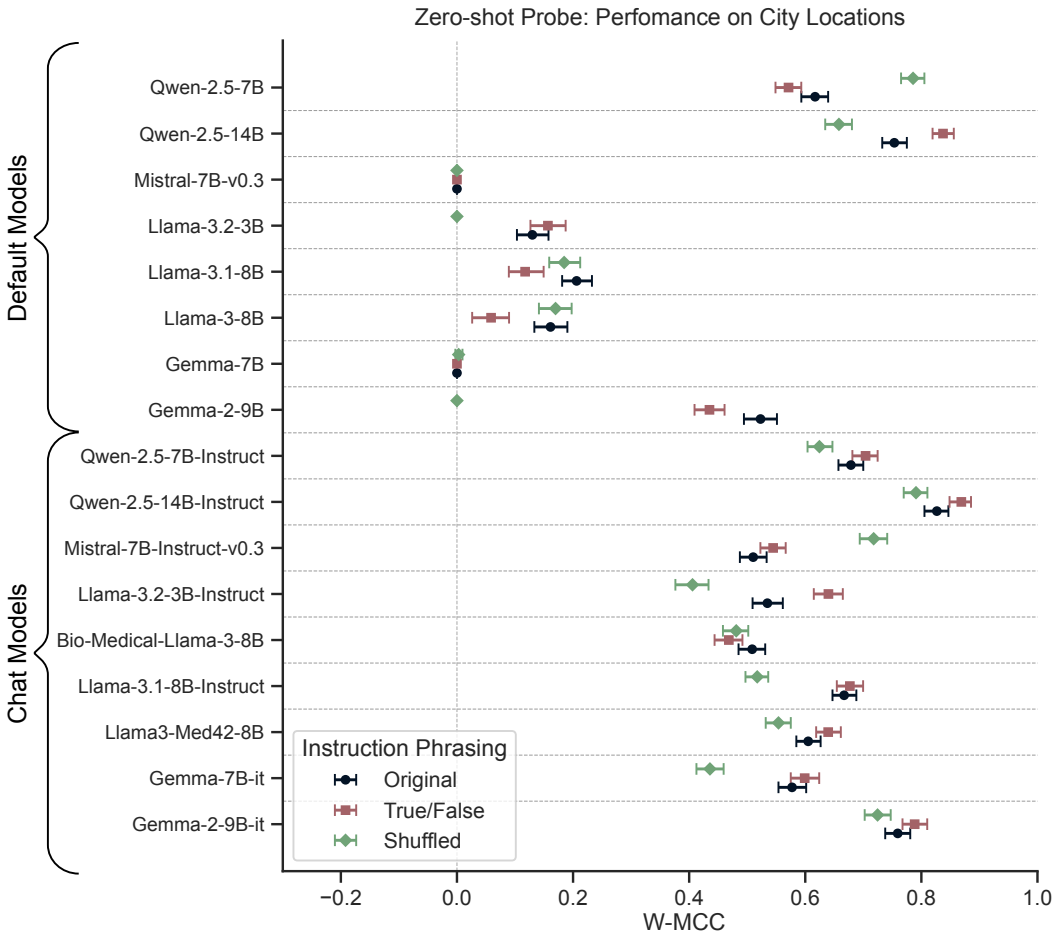

Figure 5: **Performance of zero-shot prompting on the *City Locations* data set across different models and instruction phrasings.** We use the Weighted Matthew's Correlation Coefficient (W-MCC) to quantify the performance. The marker shows the mean value and the error bars show the 95% confidence intervals (based on the bootstrapping with $n = 1{,}000$ bootstrap samples). Minimal changes to the prompt instructions can skew the performance of zero-shot prompting. Chat models exhibit the highest performance across all instruction phrasings. However, the default `Qwen` models match the performance of other chat-based models. Shuffled instructions appear to lead to worse performance in chat models. We expected that the phrasings would have only a minor effect on their performance. The default `Gemma` and `Mistral` models seem to fail (their performance is around 0).

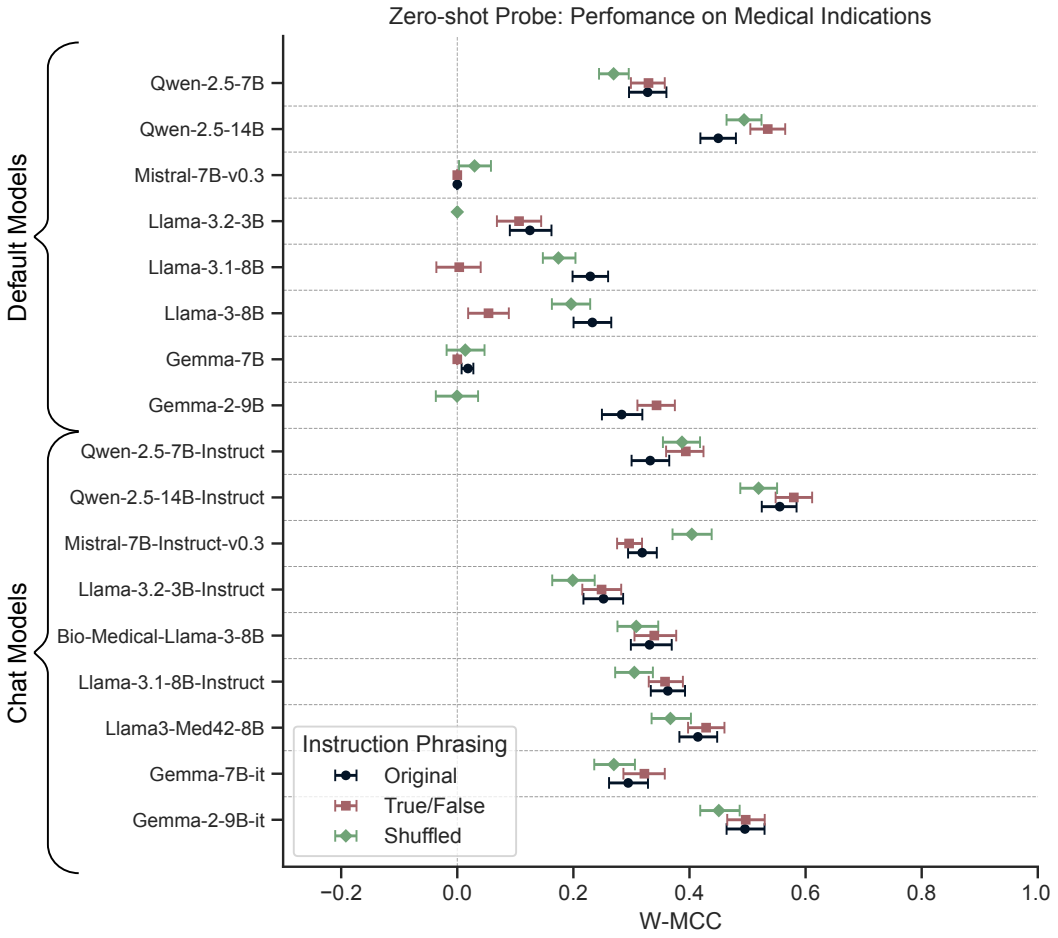

Figure 6: **Performance of zero-shot prompting on the *Medical Indications* data set across different models and instruction phrasings.** We use the Weighted Matthew's Correlation Coefficient (W-MCC) to quantify performance. The marker shows the mean value and the error bars show the 95% confidence intervals (based on the bootstrapping with $n = 1{,}000$ bootstrap samples). Minimal changes to the prompt instructions can skew the performance of zero-shot prompting. We observe a slight performance misalignment depending on the instruction phrasing. The best-performing LLMs are the largest chat models: `Gemma-2-9b` and `Qwen-2.5-14b`. We expected the biomedical `Llama` models to outperform on the medical indications dataset.

Figure 7: **Performance of the zero-shot prompting on the *Word Definitions* data set across different LLMs and instruction phrasings.** We use the Weighted Matthew's Correlation Coefficient (W-MCC) to quantify the performance: the marker shows the mean value and the error bars show the 95% confidence intervals (based on bootstrapping with $n = 1{,}000$ bootstrap samples). Minimal changes to the prompt instructions can skew the performance of zero-shot prompting. The overall performance on the *Word Definitions* data set is much lower compared to the performances on the other data sets. Generally, the misalignment in the performance between different instructions is much lower (except for the default `Qwen` models, where the difference is significant). The largest `Qwen-2.5-14b` are the top-performing models on this task.

G Conformal Prediction Intervals

In our work, we focus on "split conformal learning" [21], which requires a hold-out (or calibration) data set to compute conformal prediction intervals.

Given a probe g_i, we use a calibration data set $\mathcal{D}_{calib}$ of activation-label pairs $\langle h_i(\boldsymbol{x}), y\rangle$ to find prediction regions that ensure, for example, that a sample falling within the region is correctly classified 90% of the time. If a prediction falls into the overlapping conformal prediction intervals of two (or more) classes, or if it does not fall within any interval, the probe abstains from making any prediction. We provide pseudocode for the nonconformity functions in Supplementary Alg. 3 and 4.

G.1 Nonconformity score

To identify the conformal intervals, we compute a nonconformity score for each sample in $\mathcal{D}_{calib}$. For the binary cases (such as mean-difference probe and one-vs-all `sAwMIL`), we use the binary nonconformity scoring; see Supplementary Alg. 3. It is based on the distance between the prediction and the classifier's separating hyperplane. In Eq. 12, s is the signed distance of the sample to the separation hyperplane, and y is a ground-truth (or candidate) label:[20]

$$\textit{binaryNC}\,(s, y) = \exp\big(-y \cdot s\big),\ y \in \{-1, 1\} \text{ and } s \in \mathbb{R}. \tag{12}$$

If the sample ends up on the wrong side of the separation hyperplane (e.g., $s > 0$ and $y = -1$), then the nonconformity score in Eq. 12 is high and the candidate label is weakly supported by the model.

For the multiclass `sAwMIL`, we use the multiclass nonconformity score [29]; see Supplementary Alg. 4. For a given candidate label y, the label is defined in terms of the difference between the predicted probability of the true class and the highest probability among the other classes (with K denoting the total number of classes). Formally, for a candidate label y with predicted probability p_y, we calculate the multiclass nonconformity score with the following function:

$$\textit{multiclassNC}\,(\boldsymbol{p}) = \frac{1 - (p_y - \max_{i \neq y} p_i)}{2} \tag{13}$$

where $\boldsymbol{p} \in \Delta^{K-1} := \left\{\boldsymbol{p} \in \mathbb{R}^K \mid p_i \geq 0,\ \sum_{i=1}^{K} p_i = 1\right\}$ and Δ^{K-1} is a simplex.

In both cases, lower scores in Eq. 12 and Eq. 13 indicate that the candidate label y is strongly supported by the model. In our work, we set $\alpha = 0.1$. Thus, if the nonconformity score of a new sample exceeds the 90th quantile, the probe abstains from prediction (see Supplementary Alg. 4). The addition of conformal intervals enables us to distinguish between cases where the statements originate from different distributions, as compared to those in the calibration data set.

[20] In this case, labels should be either -1 or 1. Thus, all samples with label 0 are assigned label -1.

H More on Representation-based Probing Methods

H.1 Mean-difference Probe with Conformal Prediction Intervals

The mean-difference probe (MD+CP) consists of two components: binary mean-difference classifier (MD) and the conformal prediction intervals (CP).

First, we fit the binary classifier with a linear decision boundary [15]. We use it to separate *true* and *false* statements based on the internal activations h_i. For each pair $\langle \boldsymbol{x}_j, y_j \rangle$, we extract the activation of the last token $h_i(\boldsymbol{x}_j)_{[L]}$ and assemble a set of factually true $\mathcal{X}^+ = \{h_i(\boldsymbol{x_j})_{[L]} : y_j = \text{true}\}$ and a set of false $\mathcal{X}^- = \{h_i(\boldsymbol{x_j})_{[L]} : y_j = \text{false}\}$ activations. Here, L is the index of the last token in $\boldsymbol{x}$. We then compute the means of each set, denoted $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$, and compute a direction vector:

$$\boldsymbol{\theta} = (\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)\, \boldsymbol{\Sigma}^{-1}\, (\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)^T \tag{14}$$

In Eq. 14, $\boldsymbol{\Sigma}$ is a pooled covariance matrix. See Alg. 2 in Supplementary Materials for the detailed pseudo-code.

Second, we augment MD with conformal prediction intervals [38]. Conformal intervals help detect statements that fall outside MD's high-confidence regions for *true* or *false* classes. We use $\alpha = 0.1$ in our experiments; thus, predictions in the high-confidence regions are guaranteed to be correct at least 90% of the time. Note that we use the *true* and *false* statements from the calibration set to find the conformal prediction intervals. Finally, we test the MD+CP probe using *true*, *false*, and *neither* statements from the test set. *How can the binary MD+CP classifier identify neither statements in addition to true and false statements?* If the MD+CP probe accurately captures the veracity signal, the *neither* statements (from the test set) should fall outside of the conformal prediction intervals. Below, we observe that this is not the case. MD+CP assigns high-confidence scores to the *neither*-valued statements in the *true* or *false* regions.

Supplementary Fig. 8 shows the score distributions[21] of MD+CP on the best performing decoder (i.e., 13^{th}) of the default Llama-3-8B model on the *City Locations* data set. There are three distributions: one for *true*, one for *false*, and one for *neither*. The distributions are based on samples from the test set. If MD+CP correctly captures the veracity signal, we expect the distribution of *neither* statements (green bars) to be outside of the conformal prediction interval (i.e., in the gray area). However, this is not the case. Most of the *neither* statements fall within the conformal prediction intervals (i.e., not in the area colored gray) and get labeled as *true* or *false*.

Supplementary Fig. 9A illustrates per-token MD+CP predictions across entire statements. If MD+CP correctly identifies the veracity signal, then (1) it should not assign any labels to the tokens in the pre-actualized parts of statements $\boldsymbol{x}^p$, and (2) the label should be consistent across the actualized path $\boldsymbol{x}^a$. Given a statement "The city of Tokyo is in Japan.", the pre-actualized part is "The city of Tokyo is in" and the actualized part is "Japan." We observe that MD+CP assigns scores to the pre-actualized tokens ("The city of *X*") that fall within the conformal prediction intervals in cases #1–5 (see Supplementary Fig. 9). In case #5, MD+CP assigns a correct prediction at the *period sign* ($p = 0.39$ corresponds to a false label), but the prediction flips at the end of the text, where the *question mark* gets $p = 0.95$ corresponding to the *true* label. Similarly, in the #7 case of Supplementary Fig. 9A, the sentence does not have *any* veracity value (i.e., it is not a factual claim). However, the MD+CP probe assigns high confidence scores to some of its tokens. These findings suggest that MD+CP probe captures proxies or spurious correlations. One cannot use it in real scenarios, where we do not know a priori where the factual claim ends. In contrast, Supplementary Fig. 9 B–D show the per-token predictions for the one-vs-all sAwMIL. These probes correctly identify positions where the veracity is actualized. For example, they do not assign predictions to the non-actualized parts of the statements and only label tokens in the actualized part. Moreover, these probes do not label tokens in cases where veracity is absent (e.g., see case $\#7$ in Supplementary Fig. 9D).

H.2 Multiclass Single-Instance Support Vector Machine

The multiclass sAwMIL probe is a multiple-instance learning (MIL) version of Support Vector Machine (SVM), designed to operate on bags of token representations. To assess whether the MIL formulation

[21] We use the embedding of the last token to compute the scores in Supplementary Fig. 8: $g_i\left(h_i\left(\boldsymbol{x}\right)\right) = \boldsymbol{\theta}^T h_i(\boldsymbol{x})_{[L]} + \beta$. Here, L is the total length of the statement, which is the same as the index of the last token.

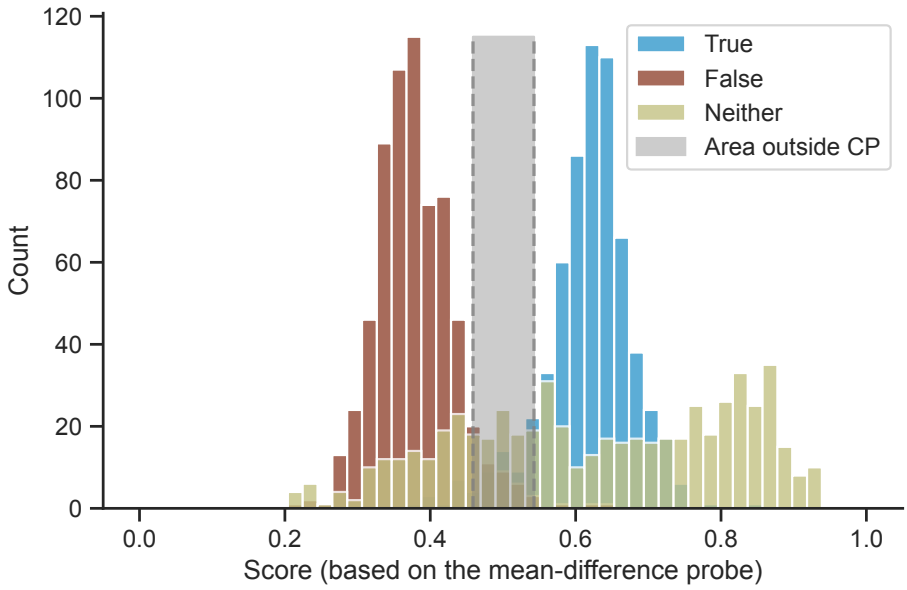

Figure 8: **Score distributions of mean-difference probe with conformal prediction intervals (MD+CP) on the 13th decoder activations of the default Llama-3-8B model for the City Locations dataset.** The probe provides a good separation between *true* and *false* statements. Note, if the MD+CP probe *truly* captures only the veracity signal, we expect the scores for the *neither* statements to fall outside the conformal intervals (i.e., be in the area highlighted with gray color). However, MD+CP assigns high-confidence scores for the *neither*-valued statements, labeling them *true* or *false*. This finding suggests that MD+CP relies on spurious proxies rather than genuine veracity signals.

offers any benefits, we construct a single-instance baseline by training a multiclass SVM on the last token representation only $h_i(\boldsymbol{x})_{[L]}$. As with multiclass sAwMIL, we first train three one-vs-all probes: is-true, is-false, and is-neither. Then, these one-vs-all classifiers are assembled into a multiclass SVM using the same procedure described in Sec. 3.1.1 and Supplementary Alg. 1. Finally, we augment the multiclass SVM with conformal prediction intervals to provide calibrated estimates, mirroring the multiclass sAwMIL setup.

As before, to evaluate performance, we provide all token representations $h_i(\boldsymbol{x})$ (not only the last one), where the final prediction is computed based on

$$\hat{g}_i(\boldsymbol{x}) \;=\; \max_{1 \le j \le L} g_i\big(h_i(\boldsymbol{x})_{[j]}\big), \text{ where } L = |\boldsymbol{x}| \text{ (number of tokens in } \boldsymbol{x}).$$

Supplementary Fig. 10 depicts the performance of multiclass sAwMIL vs. multiclass SVM. The performance of multiclass SVM is closer to the performance of multiclass sAwMIL (as compared to the performances of zero-shot prompting or MD+CP probe in Fig. 2). However, multiclass sAwMIL still outperforms the multiclass single-instance SVM in 46 out of 48 cases ($=$ 16 LLMs $\times$ 3 data sets), and is competitive in the remaining two cases. For more results, we refer the reader to Tables 10 and 12 of the Supplementary materials.

In Supplementary Fig. 10, we also observe that the multiclass sAwMIL performs better on the chat models (see bottom right portion of the plot) than the multiclass SVM. This supports our claim that veracity signals often emerge at positions other than the final token, and that multiple-instance learning can better isolate the veracity signal. Recall that the multiclass sAwMIL probe considers all the tokens in the statement and has additional training stages

Supplementary Fig. 11 visualizes the per-token predictions. The one-vs-all SVM-based probes have better selectivity than the mean-difference probe with conformal prediction intervals (MD+CP) However, in some cases, one-vs-all SVM assigns labels to the tokens in the pre-actualized part of the statement (e.g., see Supplementary Fig. 11B, #5 statement). This suggests that one-vs-all SVM probes are capturing spurious correlations with potential proxies.

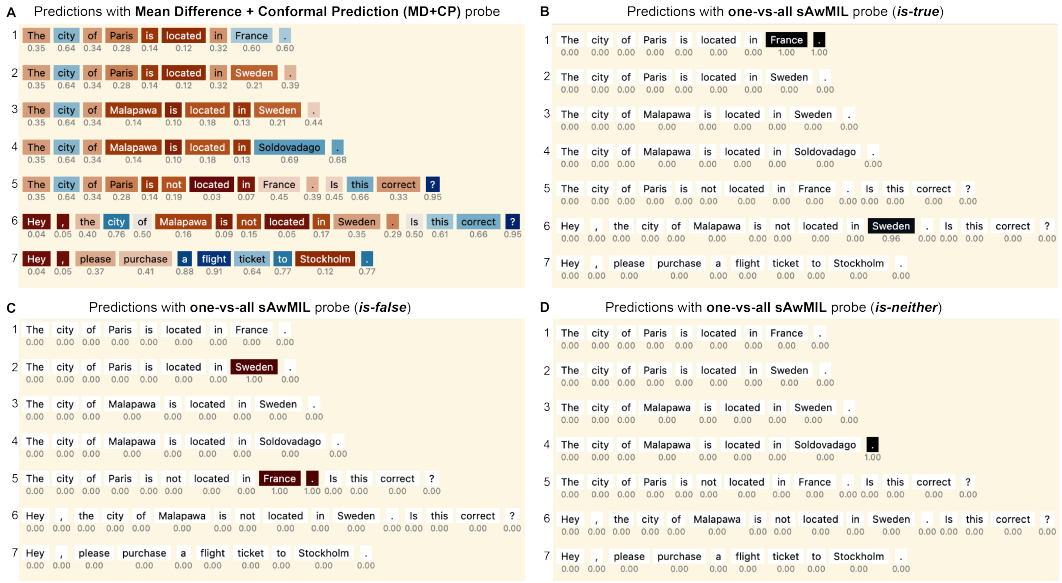

Figure 9: **Per-token predictions of mean-difference with conformal prediction intervals (`MD+CP`) and one-vs-all `sAwMIL` on the 13th decoder activations of the default `Llama-3-8B` model.** Statements are from the *City Locations* data set. We show per-token probabilities (printed beneath each word), assigned based on the token's representation. Words are shaded based on the predicted probability. When MD+CP outputs 0, the statement is labeled false; when it outputs 1, the statement is labeled true. If the per-token score falls outside the conformal intervals, `MD+CP` assigns a score of 0.5 to that token (which corresponds to the highest uncertainty). The one-vs-all `sAwMIL` probe for `is-true` outputs 1 when the probe is 100% confident that the statement is true; and it outputs 0 when the per-token score is outside the conformal intervals (i.e., there is an absence of truthfulness signal). Similarly, the one-vs-all `sAwMIL` probe for `is-false` outputs 1 when the probe is 100% confident that the statement is false; and it outputs 0 when the per-token score is outside the conformal intervals (i.e., there is an absence of falsehood signal). The same logic applies for the one-vs-all `sAwMIL` probe for `is-neither`. **Panel A** shows the `MD+CP` predictions. It often assigns high confidence scores to pre-actualization tokens and makes mistakes on the *wrapped* prompts in cases #5 and 6 (e.g., statement #6: "Hey,___ Is this correct?"). Also, `MD+CP` probe assigns labels to the statement without any veracity value (e.g., case #7). **Panels B–D** display one-vs-all `sAwMIL` probes (`is-true`, `is-false`, and `is-neither`). Unlike `MD+CP`, one-vs-all `sAwMIL` localizes the veracity signal to the actualized token and abstains elsewhere, demonstrating superior selectivity.

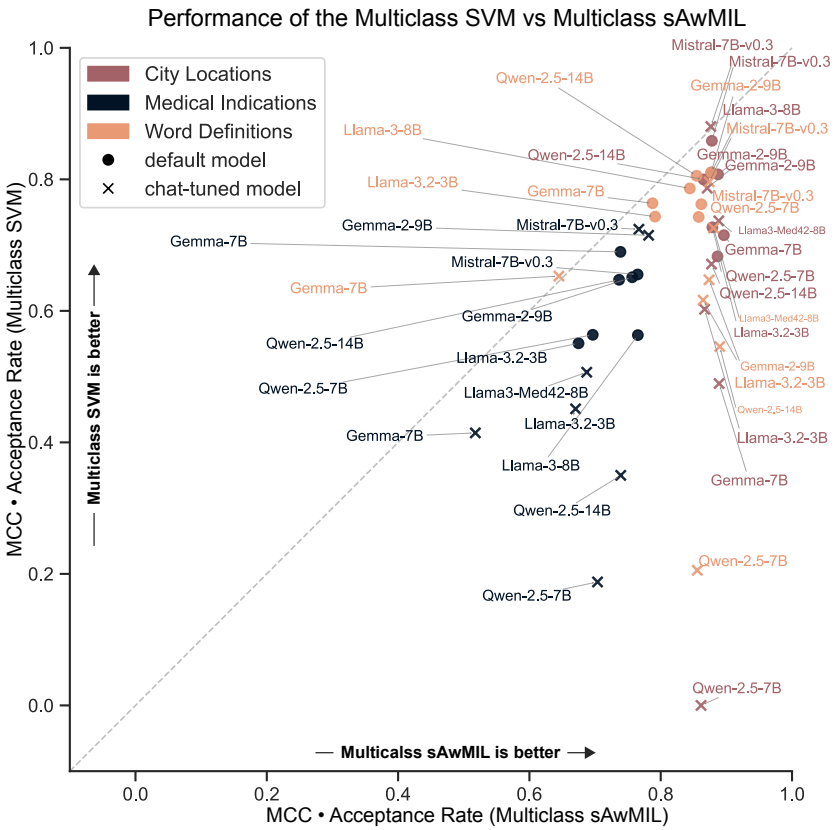

Figure 10: **Performance of multiclass multiple-instance SVM (i.e., multiclass `sAwMIL`) vs. multiclass single-instance SVM probes.** Each marker shows a probe's performance for a ⟨model, dataset⟩ pair. Default models are shown with circles, while chat models are shown with crosses. The different colors indicate the different data sets. The performance of the multiclass `sAwMIL` probe is specified on the x-axis, while the performance of the multiclass SVM is specified on the y-axis. As before, we report the Weighted Matthew's Correlation Coefficient (W-MCC). Multiclass SVM is trained on the representation of the last token only. The evaluation is the same for both probes and is based on all tokens in the statement. That is, to prediction, each probe takes the maximum score (across all the tokens in the statement). We observe that multiclass `sAwMIL` outperforms multiclass SVM. The only exceptions are the `Gemma-7B` chat model and the `Mistral-7B-v0.3` chat model on *Word Definitions*, where the performances of the single-instance and multiple-instance are competitive. This experiment shows that multi-instance learning (i.e., training on all the tokens in the statement) is beneficial when tracking the veracity of an LLM.

Figure 11: **Per-token predictions of one-vs-all single-instance SVM on the 13th decoder activations of the default** `Llama-3-8B` . Statements are from the *City Locations* data set. We show per-token probabilities (printed beneath each word), assigned based on the token's representation. Words are shaded based on their predicted probability. **Panels A–C:** display the one-vs-all `SVM` probes (`is-true`, `is-false`, and `is-neither`). The one-vs-all SVM probe isolates the signal better than the `MD+CP` probe. (See Supplementary Fig. 9A for the `MD+CP` results.) However, in some instances, the one-vs-all SVM probe assigns high-certainty scores to tokens that do not have any veracity signal. For example, in Panel A (case #6, the probe picks up on tokens including 'this,' 'is,' and '?', which do not have inherent veracity value. Overall, the multiclass `sAwMIL` in Supplementary Fig. 9B–C has better selectivity.

I Additional Evaluation Details

In the Supplementary Sec. E, we provide the full list of the validity criteria for the. In the manuscript, we only cover the results related to **Correlation** and **Locality** criteria (see Sec. 5). Here, we provide additional details behind the evaluation of the **Correlation** and **Locality** (see below in Supplementary Sec. I.1.1) Further, we describe the experimental setup related to the Generalization (also see below in Supplementary Sec. I.1.1), and setups for the **Manipulation** and **Locality** criteria (see Supplementary Sec. I.1.2).

Finally, we provide the evaluation results for the **Generalization**, **Manipulation** and **Locality** in Supplementary Sec. I.2

I.1 Evaluation Setup

I.1.1 Performance and Validity

Here, we describe a pipeline to evaluate our `sAwMIL` probe over the validity criteria specified in Sec. 3 and Supplementary Sec. E.

Correlation and Selectivity. We use the test split of each data set to evaluate the performance of the probe. We use Matthew's Correlation Coefficient (MCC) to summarize the statistical accuracy of probes. The multiclass MCC value is calculated using Eq. 15.

$$\mathrm{MCC} = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{\left(s^2 - \sum_k^K p_k^2\right) \times \left(s^2 - \sum_k^K t_k^2\right)}}, \tag{15}$$

where c is the number of correct predictions, s is the total number of samples, K is the total number of classes, t_k is the number of k-class samples in the data set, and p_k is the number of times k-class was predicted. $\mathrm{MCC} = 1$ indicates that a classifier predicted every instance correctly. $\mathrm{MCC} = 0$ implies that the predictions are random. $\mathrm{MCC} = -1$ indicates that the predictions are inversely correlated with the ground-truth labels.

Zero-shot prompting, one-vs-all `sAwMIL`, and multiclass `sAwMIL` can abstain from making predictions. If a probe abstains too often, it suggests poor performance. For these cases, we use Weighted-MCC (W-MCC), where the *acceptance rate* serves as the weight (see Eq. 16).

$$\text{W-MCC} = \mathrm{MCC} \times \left(1 - \frac{\text{\# abstained}}{\text{\# total predictions}}\right) \tag{16}$$

In contrast to the other probing methods, the `MD+CP` probe cannot abstain. When a statement $\boldsymbol{x}$ is too unusual, the probe labels it as *neither*. For this case, we use the MCC score.

Since *neither* statements are included in the test data set, W-MCC and MCC provide a sense of how well the probe classifies factually *true* or *false* statements and indicate whether the probes can handle *neither*-type cases.

Generalizability. To test how well a particular probe g_i trained on data set $\mathcal{D}_i$ generalizes, we evaluate its performance using the test split of other data set $\mathcal{D}_j$—e.g., g_i trained on the city locations data set is evaluated using the test split of the word definitions data set.

I.1.2 Interventions and Validity

In this experiment, we assess whether perturbing the hidden representation $h_i(\boldsymbol{x})$ along the veracity direction $\vec{\nu}_i$ affects the model's outputs, and whether these interventions satisfy the manipulation and locality criteria defined in Supplementary Sec. E. In other words, we use $\vec{\nu}_i$ to change the distribution of the output tokens $P_{\mathcal{M}}$ and force true or false responses.

We look at each factually true statement $\boldsymbol{x} \in \mathcal{D}_{test}$—e.g., "The city of Santo Domingo is in the Dominican Republic." We split these statements into two segments: a pre-actualized part, $\boldsymbol{x}^p$, such as "The city of Santo Domingo is in"; and second, an actualized part, $\boldsymbol{x}^a$ such as "the Dominican Republic". Given $\boldsymbol{x}^p$, we can compute the probability of the actualized part according to Eq. 17:

$$P_{\mathcal{M}}(\boldsymbol{x}^a_{[1:L]} \mid \boldsymbol{x}^p) = \prod_{l=1}^{L} P_{\mathcal{M}}\left(\boldsymbol{x}_{[l]} \mid \boldsymbol{x}^a_{[0:(l-1)]}, \boldsymbol{x}^p\right). \tag{17}$$

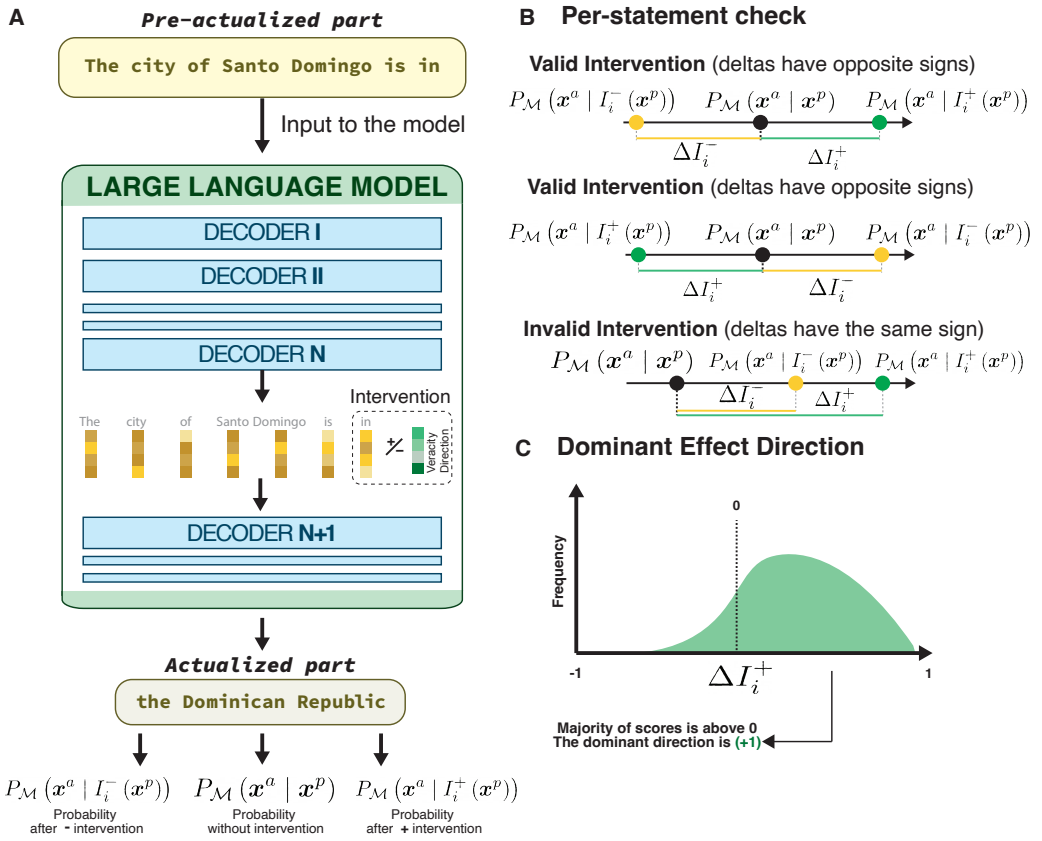

Figure 12: **Workflow for evaluating the success of directional interventions.** (**Panel A**) We provide a pre-actualized part $\boldsymbol{x}^p$ to the LLM, and intervene at the final token embedding after the n-th decoder by adding or subtracting the learned veracity direction $\vec{\nu}_i$. The modified representation is then passed to decoder $n+1$. Further, we compute the conditional probability of the *correct* actualized part $\boldsymbol{x}^a$. For each statement, we obtain three values: the original conditional probability, the probability after a positive shift $+\vec{\nu}_i$, and the probability after a negative shift $-\vec{\nu}_i$. (**Panel B**) We assess the success of each intervention at the statement level, comparing the change in conditional probabilities. If the change is caused by the positive shift (ΔI_i^+) and the negative shift (ΔI_i^-) are of opposite sign, we consider the intervention to have a consistent directional effect and mark it as successful for that statement (see Eq. 21). (**Panel C**) To evaluate the success of the intervention, we also identify the *dominant effect direction*—i.e., the sign of ΔI_i^+ that appears most frequently across all statements. The procedures in panels B and C determine the per-statement success (see Eq. 21). If more than half of the statements show consistent and directionally aligned changes (see Eq. 22), we consider the overall intervention along $\vec{\nu}_i$ to be successful.

In Eq. 17, L is the number of tokens in the actualized part $\boldsymbol{x}^a$.[22] The subscript $[l]$ specifies the index of a token in $\boldsymbol{x}^a$, and $[1:l]$ specifies the range of tokens in $\boldsymbol{x}^a$, while $\boldsymbol{x}^a_{[0]}$ refers to an empty set. Further, we do not specify a subscript $[1:L]$ for brevity unless it is necessary for clarity.

Direction vector $\vec{v}_i$. We train one-vs-all `sAwMIL` probes via dual optimization. We use the obtained solution to extract the linear direction that points towards the class of interest. For example, in our `is-true` probe, the class of interest is the true statements.

$$\vec{v}_i = \sum_{j \in \mathcal{S}} \alpha_j \, y_j \, h_i(\boldsymbol{x}_j) \;\; \text{with } \mathcal{S} = \{j \mid \alpha_j > 0\},\; a_j \in \mathbb{R},\; y_j \in \{-1, 1\}. \tag{18}$$

[22] The length of the actualized part depends on the tokenization technique the language model uses. For example, `Llama-3` and `Mistral` models split *Albania* into `[Al] [ban] [ia]`, while `Gemma` models have one reserved token `[Albania]`.

In Eq. 18, $\mathcal{S}$ is a set of support vectors, α_j is the Lagrangian multiplier [39], $h_i(\boldsymbol{x}_j)$ is the activation after the i^{th} decoder for statement $\boldsymbol{x}_j$, and y_j is the class label for $\boldsymbol{x}_j$.

Interventions. Given a pre-actualized part of the statement $\boldsymbol{x}^p$, model $\mathcal{M}$, and a hidden representation $h_i(\boldsymbol{x}^p)$, we apply directional interventions by translating the representation of the last token $\boldsymbol{x}^p_{[L]}$ along $\pm\,\vec{\nu}_i$ (here, i stands for the index of the decoder):

- **Positive directional shift**:

$$I_i^+\left(h_i\left(\boldsymbol{x}^p\right)\right) = \left[h_i(\boldsymbol{x}^p)_{[1]}, \ldots, h_i(\boldsymbol{x}^p)_{[L-1]}, h_i(\boldsymbol{x}^p)_{[L]} + \vec{\nu}_i\right]$$

- **Negative directional shift**:

$$I_i^-\left(h_i\left(\boldsymbol{x}^p\right)\right) = \left[h_i(\boldsymbol{x}^p)_{[1]}, \ldots, h_i(\boldsymbol{x}^p)_{[L-1]}, h_i(\boldsymbol{x}^p)_{[L]} - \vec{\nu}_i\right]$$

These interventions return modified representations, which we denote as $I_i^+(\boldsymbol{x}^p)$ and $I_i^-(\boldsymbol{x}^p)$, respectively. Furthermore, we compute the per-sample effect of the directional interventions. It is defined as a difference between the original probability and the probability we get after the intervention:

$$\Delta I_i^+\left(\boldsymbol{x}^a, \boldsymbol{x}^p\right) \leftarrow P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid I_i^+\left(\boldsymbol{x}^p\right)\right) - P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid \boldsymbol{x}^p\right) \tag{19}$$

$$\Delta I_i^-\left(\boldsymbol{x}^a, \boldsymbol{x}^p\right) \leftarrow P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid I_i^-\left(\boldsymbol{x}^p\right)\right) - P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid \boldsymbol{x}^p\right) \tag{20}$$

To compare interventions across decoders and models, we look at the success rate of the interventions. The per-statement intervention is successful if ΔI_i^+ and ΔI_i^- have opposing effects on the conditional probability $P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid \boldsymbol{x}^p\right)$. In other words, if ΔI_i^+ is positive, then ΔI_i^- must be negative, and vice-versa.

At the same time, we must ensure that the effect of the intervention is consistent in most statements $\boldsymbol{x} \in \mathcal{D}_{test}$. If half of the statements have positive ΔI_i^+ and another have negative ΔI_i^+, then our intervention produces a random change in $P_{\mathcal{M}}$. To ensure that the effect is consistent, we look at the *dominant effect direction* of the intervention, $\overline{d}_i \in \{-1, +1\}$. Here, $\overline{d}_i = +1$, if more than half ΔI_i^+ are positive, and $\overline{d}_i = -1$ if more than half is negative.

In summary, a successful directional intervention is one that produces opposing effects when shifting along $\pm\vec{\nu}_i$, and aligns ΔI_i^+ with the dominant direction. Supplementary Fig. 12 provides an overview of the workflow to determine the per-statement success of the intervention. Formally, we define a per-statement success s as

$$s(\boldsymbol{x}) = \mathbb{I}\Big[\left[\operatorname{sign}\left(\Delta I_i^+\left(\boldsymbol{x}^a, \boldsymbol{x}^p\right)\right) \neq \operatorname{sign}\left(\Delta I_i^-(\boldsymbol{x}^a, \boldsymbol{x}^p)\right)\right] \wedge \left[\operatorname{sign}\left(\Delta I_i^+\left(\boldsymbol{x}^a, \boldsymbol{x}^p\right)\right) = \operatorname{sign}\left(\overline{d}_i\right)\right]\Big] \tag{21}$$

where

$$\mathbb{I}\left[\cdot\right] = \begin{cases} 1, & \text{if the condition holds,} \\ 0, & \text{otherwise.} \end{cases}$$

Why do we only look at the sign? During our initial experiments, we observed that even when $\vec{\nu}_i$ was trained to separate true and false statements, the effect of the intervention on $P_{\mathcal{M}}(\boldsymbol{x}^a \mid \boldsymbol{x}^p)$ could vary across decoders. Specifically, in certain decoders, shifting along $\pm\vec{\nu}_i$ consistently increased the probability of $\boldsymbol{x}^a$, while in others it decreased it. These effects were consistent in the sense that shifting in the opposite direction produced the opposing effect. This phenomenon can be attributed to the complex interactions within each decoder of the model. As highlighted by Heimersheim and Nanda [40], activation patching experiments have revealed that interventions can have varying directions of effect depending on the decoder. Therefore, observing a sign flip in the effect of an intervention does not invalidate the direction $\vec{\nu}_i$, as long as it is consistent.

Given a set of true statements from $\mathcal{D}_{test}$, we compute the overall success rate at a decoder i as follows:

$$\omega_i \;=\; \frac{1}{N}\sum s\left(\boldsymbol{x}_j\right), \tag{22}$$

where N stands for the number of the true statements in $\mathcal{D}_{test}$ and $s\left(\boldsymbol{x}_j\right)$ is success for the j^{th} statement as defined in Eq. 21.

If the overall success rate at decoder i is greater than 50%, we claim that the intervention at decoder i is successful. Hence, the manipulation criterion is fulfilled. We use a one-sided binomial test to confirm whether the overall success rate is significant:

$$H_0 : \omega_i \leq 0.5 \tag{23}$$

$$H_A : \omega_i > 0.5 \tag{24}$$

For example, the overall success rate of 61% with the dominant direction $\overline{d}_i = +1$ tells us that if we have 100 statements, on average, we increase the probability of the correct answer in 61 statements. Similarly, a success rate of 98% with the dominant direction $\overline{d}_i = -1$ indicates that shifting along $+\vec{\nu}_i$ decreases the probability of the correct answer in approximately 98 out of 100 statements.

Locality. To further determine the quality of the directional intervention, we assess whether changes in probability are concentrated on the actualized part, rather than being diffused across random tokens in the vocabulary $\mathcal{V}$. In other words, our intervention should change $P_{\mathcal{M}}(\boldsymbol{x}^a \mid \boldsymbol{x}^p)$ and should not change the probability of random tokens $P_{\mathcal{M}}(\boldsymbol{r} \mid \boldsymbol{x}^p)$.

Specifically, we expect the intervention to primarily affect the likelihood of the correct continuation, $P_{\mathcal{M}}(\boldsymbol{x}^a_{[1:L]} \mid \boldsymbol{x}^p)$, while leaving the probability of a randomly sampled continuation, $P_{\mathcal{M}}(\boldsymbol{r}_{[1:L]} \mid \boldsymbol{x}^p)$, mostly unchanged. Here, $\boldsymbol{r}_{[1:L]}$ denotes a random sequence sampled from the vocabulary of the model $\mathcal{M}$. We quantify these changes as:

$$\Delta_{\text{Correct}} = \left| P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid I_i^+(\boldsymbol{x}^p)\right) - P_{\mathcal{M}}\left(\boldsymbol{x}^a \mid I_i^-(\boldsymbol{x}^p)\right) \right| \tag{25}$$

$$\Delta_{\text{Random}} = \left| P_{\mathcal{M}}\left(\boldsymbol{r} \mid I_i^+(\boldsymbol{x}^p)\right) - P_{\mathcal{M}}\left(\boldsymbol{r} \mid I_i^-(\boldsymbol{x}^p)\right) \right| \tag{26}$$

Further, we say that the intervention satisfies the *locality* criterion if

$$\mathbb{E}\left[\Delta_{\text{Correct}}\right] \; > \; \mathbb{E}\left[\Delta_{\text{Random}}\right]. \tag{27}$$

That is, the expected change in probability for the correct output, $\boldsymbol{x}^a$, exceeds the expected change for a randomly sampled output $\boldsymbol{r}$.

I.2 Results

I.3 Generalization Across Data Sets

To further support the claim that the multiclass `sAwMIL` captures veracity signals (and not merely a proxy), we demonstrate generalization performance across data sets. Supplementary Fig. 13 provides results for each data set and LLM. The columns correspond to three test data sets, and the cells specify the multiclass `sAwMIL`'s performance for a specific LLM. Multiclass `sAwMIL` provides reasonable generalization performance (see Supplementary Tab. 8). However, it is potentially overfitting to the highly specialized *City Locations* data set. Using more diverse data sets that contain a broader range of entities and cover a larger set of topics isolates the veracity signal better and produces better generalization performance. We refer the reader to Tables 13 through 15 in Supplementary for detailed statistics.

Table 8: **Aggregated generalization performance of the multiclass `sAwMIL` for each dataset.** Each cell shows a MCC value, which quantifies the performance of the multiclass `sAwMIL` trained and tested on different combinations of the datasets. The value in the bracket is the standard error. *Word Definitions* provides better generalization performance because it contains statements covering a diverse set of topics, while the *City Locations* provide lower generalization performance.

	Testing Dataset		
Training Dataset	City Locations	Medical Indications	Word Definitions
City Locations	0.963 (0.003)	0.624 (0.030)	0.633 (0.025)
Medical Indications	0.818 (0.033)	0.790 (0.009)	0.698 (0.018)
Word Definitions	0.896 (0.015)	0.723 (0.016)	0.868 (0.008)

We also observe that generalization performance is higher for chat models, where the average MCC score (on the non-training data set) is 77.2% (standard error: 0.2%), compared to 68.2% (standard

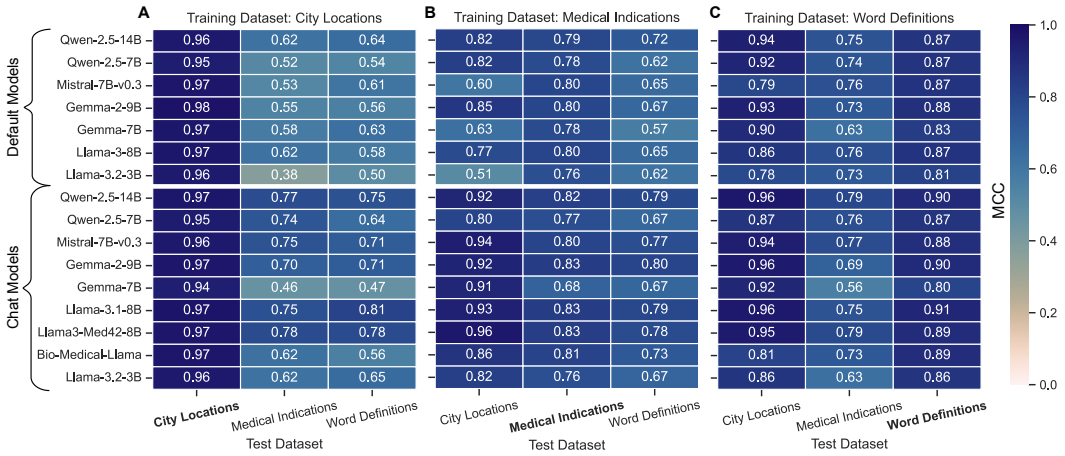

Figure 13: **Generalization performance of the multiclass `sAwMIL` probe across data sets.** Each panel corresponds to a different *training* data set: *City Locations*, *Medical Indications*, and *Word Definitions*. Each column corresponds to a different test data set. Each cell displays MCC values, which quantify how well the probe generalizes to the test data set (a higher value is better). For each model and data set, we report the maximum MCC achieved across all decoders. Generally, probes trained on the chat models have better generalization performance than the default models. **Panel A:** Generalization performance of multiclass `sAwMIL` when trained on *City Locations*. While the MCC values are significantly higher than random baseline (with MCC = 0), the generalization ability is lower than those in Panels B or C. **Panel B:** Generalization performance of multiclass `sAwMIL` when trained on *Medical Indications*. In Panel A, we observe that training on *City Locations* and testing on *Medical Indications* provides good but not excellent MCC values (average MCC of 0.624 with standard error of 0.030). This is not the case in this panel, where *Medical Indications* is the training data set and *City Locations* is the test data set (average MCC of 0.818 with standard error of 0.033). **Panel C:** Generalization performance of multiclass `sAwMIL` when trained on *Word Definitions*. This probe has high generalization performance across data sets. When *City Locations* is the test data set, the average MCC is 0.896 with standard error of 0.015; and when *Medical Indications* is the test data set, the average MCC is 0.723 with standard deviation of 0.016. For aggregated statistics, see Supplementary Tab. 8.

error: 0.2%) achieved for the default models. This is more noticeable in Supplementary Fig. 13A, where default models have much lower MCC values than their chat model counterparts. For example, the chat model `Llama-3.2-3B` has 1.6 times higher MCC value than the default `Llama-3.2-3B`. Over the three panels, `Gemma-7B` seems to be an outlier, since the generalization performance drops significantly for the chat version of the model.

Multiclass `sAwMIL` satisfies the generalization criterion defined in Supplementary Sec. E by transferring veracity probes trained on one data set to another while maintaining strong performance. This provides further evidence that multiclass `sAwMIL` captures a veracity signal that is not specific to a data set.

I.4 Interventions: Manipulation and Locality

Previous experiments have shown that the multiclass `sAwMIL` identifies a strong and transferable veracity signal. We further look at how this signal is connected to the output of an LLM, $P_{\mathcal{M}}$. Here, we look at the interventions of one-vs-all `sAwMIL` probes, which are the building blocks of the multiclass `sAwMIL`. Specifically, we assess the effectiveness of interventions applied along the `is-true` and `is-false` directions. For simplicity, we exclude the one-vs-all probe for `is-neither` from the intervention analysis.

The overall success rate measures how often directional interventions (adding or subtracting $\pm\vec{\nu}_i$) produce a consistent change in the model's output $P_{\mathcal{M}}$. We describe the setup in Supplementary Sec. I.1.2. A higher success rate indicates that shifting along $\pm\vec{\nu}_i$ has higher chances to skew the conditional probability of the correct answers $P_{\mathcal{M}}(\boldsymbol{x}^a \mid \boldsymbol{x}^p)$. Supplementary Fig. 14 shows the

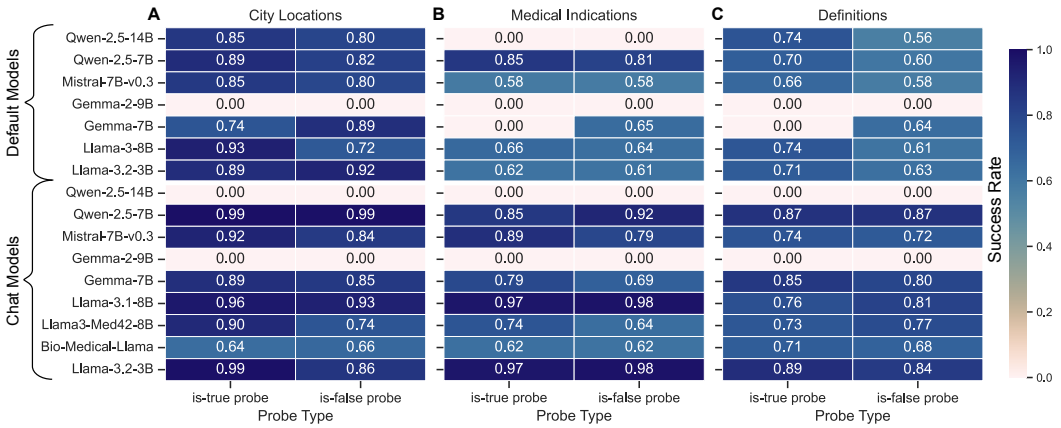

Figure 14: **Overall intervention success rate for the one-vs-all `sAwMIL` probes.** We report the maximum achievable success rate along the `is-true` and `is-false` directions. Panels **A**-**C** show results for probes g_i trained on specific data sets, while columns correspond to the `is-true` or `is-false` probe. Default and chat `Gemma-2-9b`, as well as, some experiments with the default and chat `Qwen-2.5-14B` and `Gemma-7B` models did not pass the consistency check – suggesting that the interaction between directions $\vec{\nu}_i$ and the conditional probability $P_{\mathcal{M}}$ are not linear (or that the one-vs-all `sAwMIL` probes failed to identify signals that linearly affect the model output $P_{\mathcal{M}}$.

overall success rate for each model and data set. Success rates below .5 suggest that interventions have close to a random effect on $P_{\mathcal{M}}$.

Notably, some models have a success rate of 0. This occurs when:

- Interventions along $\pm\vec{\nu}_i$ failed to induce opposing changes in $P_{\mathcal{M}}$, e.g., both $+\vec{\nu}_i$ and $-\vec{\nu}_i$ increased or decreased probabilities; or
- The average change in $P_{\mathcal{M}}(\boldsymbol{x}^a \mid \boldsymbol{x}^p)$ (i.e., Δ_{correct}) matched the change in random sequence continuation probabilities (Δ_{random}), violating the locality criterion in Equations 25 through 27.

The average success rate is 80.1% (standard error: 0.2%) for the `is-true` direction and 76.2% (standard error: 0.2%) for the `is-false` direction. We exclude the models whose intervention success rate was 0 in Supplementary Fig. 14.

This experiment shows that in the majority of cases, we can use the `is-true` and `is-false` directions to manipulate the output of LLMs. The interventions are more successful for the chat models. The average success rate for chat models is 80.1% (standard error: 0.3%), and for default models is 70.9% (standard error: 0.2%).

Anomalies. We observe some anomalous behavior when intervening in the LLMs:

- In the `Gemma-2-9B` models, interventions along the direction $\pm\vec{\nu}_i$ consistently increased or decreased the probability $P_{\mathcal{M}}(\boldsymbol{x}^a \mid \boldsymbol{x}^p)$, regardless of the sign of the intervention. This indicates that the direction $\vec{\nu}_i$ identified by `sAwMIL` does not have a clear relationship with the model's output probabilities. Thus, we cannot use $\vec{\nu}_i$ to increase and decrease the probability of correct answers. A similar phenomenon is observed in the `Qwen-2.5-14B` (chat) model.
- For the `Gemma-7B` (default) model, interventions along the `is-false` direction provide a higher success rate compared to the `is-true` direction.

The exact reasons for these anomalies are unclear. However, we know that these models have additional fine-tuning processes. These additional training procedures may have influenced the internal representations of the models. Except for the `Gemma-2-9B` models and the `Qwen-2.5-14B` (chat) model, the one-vs-all `sAwMIL` probes pass both manipulation and locality criteria. We expect that the non-linear version of `sAwMIL` will overcome the issues with the models that have additional fine-tuning processes. This is part of our future work.

I.5 Recap: Overall Validity

In this section and Sec. 5 of the manuscript, we established that the multiclass `sAwMIL` probe satisfies the validity criteria. Specifically, we confirmed that it satisfies the correlation and selectivity criteria, outperforming zero-shot prompting and mean-difference probe with conformal prediction intervals. We further demonstrated that multiclass `sAwMIL` satisfies the generalization criterion, indicating that we can successfully apply probes trained on multiclass `sAwMIL` to statements from other domains. In addition, we showed that the one-vs-all `sAwMIL` probes satisfy the manipulation and locality criteria. In a majority of cases, we can perform interventions that change the probabilities of correct replies. Together, these findings provide strong evidence for the overall validity of `sAwMIL` probes. Not all LLMs have a veracity mechanism that has a linear relationship with the output. Exploring this non-linear relationship is part of our future work.

J Algorithms

In this section, we provide pseudo-codes for several procedures described in the main text. In Supplementary Alg. 1 provides pseudo-code for the Sparse Aware Multiple-Instance Learning (`sAwMIL`) probe, and Supplementary Alg. 2, we show pseudo-code for the mean-difference (`MD`) probe. In Supplementary Algorithms 3 and 4, we describe the procedure for the binary and multiclass conformal learning (described in Sec. 3.2).

Algorithm 1 Training a one-vs-all `sAwMIL` classifier

Input: A training data set $\{(\boldsymbol{x}_i, y_i, \boldsymbol{m}_i)\}_{i=1}^{n}$ with binary bag labels $y_i \in \{0, 1\}$, bags $\boldsymbol{x}_i \in \mathbb{R}^{L_i \times d}$, and intra-bag confidences $\boldsymbol{m}_i \in \{0, 1\}^{L_i}$, where L_i is the number of items in a bag $\boldsymbol{x}_i$; also, a balancing parameter $\eta \in (0, 1]$.

Output: Parameters $\boldsymbol{\theta} \in \mathbb{R}^{1 \times d}$ and $b \in \mathbb{R}$ for a linear probe g_i.

1: Partition data into positive and negative sets:

$$\mathcal{X}^+ = \{\langle \boldsymbol{x}_i, \boldsymbol{m}_i \rangle : y_i = 1\}, \quad \mathcal{X}^- = \{\langle \boldsymbol{x}_i, \boldsymbol{m}_i \rangle : y_i = 0\}$$

2: Compute the **initial** coefficient vector and the intercept ▷ See Bunescu and Mooney [19]

$$(\hat{\boldsymbol{\theta}}, \hat{b}) \leftarrow \texttt{solve\_sMIL}(\mathcal{X}^+, \mathcal{X}^-), \text{ where } \hat{\boldsymbol{\theta}} \in \mathbb{R}^{1 \times d} \text{ and } \hat{b} \in \mathbb{R}.$$

3: Let $\bar{\mathcal{X}}^+$ denote the set of all instances from the positive bags and $\bar{\mathcal{X}}^-$ all instances from the negative bag.

4: Compute scores for every instance in a positive set

$$S^+ \leftarrow \bar{\mathcal{X}}^+ \hat{\boldsymbol{\theta}}^T + \hat{b}, \text{ where } \bar{\mathcal{X}}^+ \in \mathbb{R}^{|\bar{\mathcal{X}}^+| \times d}.$$

5: Compute the threshold

$$q \leftarrow \texttt{quantile}(\mathcal{S}^+, 1 - \eta).$$

6: **for all** positive instances $\langle \bar{\boldsymbol{x}}_j, \bar{m}_j, \bar{y}_j \rangle \in \bar{\mathcal{X}}^+$, where $\bar{\boldsymbol{x}}_j \in \mathbb{R}^{1 \times d}$, $\bar{m}_j \in \{0, 1\}$ and $\bar{y}_j = \emptyset$ **do**
  if $(\bar{\boldsymbol{x}}_j \hat{\boldsymbol{\theta}}^T + b) \geq q_\eta$ and $\bar{m}_j = 1$ **then** set $\bar{y}_j = 1$;
  else set $\bar{y}_j = 0$.

7: **end for**

8: Compute the **final** coefficient vector and the intercept ▷ via simple support vector machine

$$(\boldsymbol{\theta}, b) \leftarrow \texttt{solve\_SIL}(\bar{\mathcal{X}}^+, \bar{\mathcal{X}}^-).$$

return $\boldsymbol{\theta} \in \mathbb{R}^{1 \times d}$, $b \in \mathbb{R}$.

Algorithm 2 Training a mean-difference (`MD`) probe, sometimes referred to as mean-mass or mean-cluster difference classifier/probe/separator.

Input: A training dataset $\{\langle \boldsymbol{z}_i, y_i \rangle\}_{i=1}^n$ with binary labels $y_i \in \{0, 1\}$ and $\boldsymbol{z}_i \in \mathbb{R}^{1\times d}$. In our experiments, $\boldsymbol{z}$ is the embedding of the last token (unless otherwise noted).

Output: Parameters $\boldsymbol{\theta} \in \mathbb{R}^{1\times d}$, $\beta \in \mathbb{R}$, and $\boldsymbol{\Sigma}^{-1} \in \mathbb{R}^{d\times d}$. These parameters are subsequently given to a function f along with $\boldsymbol{z}$ to compute $f(\boldsymbol{z}) = \sigma\big(\boldsymbol{z}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta}^T\right) + b\big)$, where σ is a sigmoid function.

Partition data into positive and negative sets:

$$\mathcal{X}^+ = \{\boldsymbol{z}_i : y_i = 1\}, \quad \mathcal{X}^- = \{\boldsymbol{z}_i : y_i = 0\}$$

Compute class means $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$, and covariance matrices $\boldsymbol{\Sigma}^+$ and $\boldsymbol{\Sigma}^-$ for $\mathcal{X}^+$ and $\mathcal{X}^-$.

Compute pooled covariance matrix (where $n^+ = |\mathcal{X}^+|$ and $n^- = |\mathcal{X}^-|$):

$$\boldsymbol{\Sigma} = \frac{(n^+ - 1)\,\boldsymbol{\Sigma}^+ + (n^- - 1)\,\boldsymbol{\Sigma}^-}{n^+ + n^- - 2}$$

Compute the coefficient vector: $\boldsymbol{\theta} = \boldsymbol{\mu}^+ - \boldsymbol{\mu}^-$, where $\boldsymbol{\theta} \in \mathbb{R}^{1\times d}$.

Compute scores for positive and negative sets:

$$s^+ \leftarrow \mathcal{X}^+\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta}^T\right) \text{ and } s^- \leftarrow \mathcal{X}^-\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta}^T\right), \text{ where } s^+ \in \mathbb{R}^{n^+} \text{ and } s^- \in \mathbb{R}^{n^-}.$$

Compute the intercept:

$$b = \frac{1}{2}\left(\text{mean}\left(s^+\right) + \text{mean}\left(s^-\right)\right)$$

return Coefficient vector $\boldsymbol{\theta}$, intercept β, and the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$.

Algorithm 3 Inductive Conformal Predictions with binary nonconformity score.

Input: A calibration dataset $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^n \subseteq \mathcal{D}_{cal}$, where $n = |\mathcal{D}_{cal}|$, $y_i \in \{-1, 1\}$ and $\boldsymbol{z}_i \in \mathbb{R}^{L\times d}$ ($L = 1$ in a single instance setup); a confidence level α, a pretrained binary classifier g, and a new sample $\boldsymbol{z}_{new}$.

Output: Prediction set $\mathcal{Y}_{new}$

Initialize an empty score list $S \leftarrow \emptyset$

for all samples $\langle \boldsymbol{z}_i, y_i \rangle \in \mathcal{D}_{cal}$ **do**
    $z_i = g\left(\boldsymbol{z}_i\right)$ ▷ $z_i \in \mathbb{R}$ is a score
    $s_i = \exp(-y_i \cdot z_i)$ ▷ Nonconformity score, see Eq. 12.
    $S \leftarrow S \cup \{\, s_i\}$
end for

Compute a score for the new samples: $z_{new} = g\left(\boldsymbol{z}_{new}\right)$

Initialize empty prediction set $\mathcal{Y}_{new} \leftarrow \emptyset$

for all $y \in \{-1, 1\}$ **do**
    $s_{new} = \exp(-y \cdot z_{new})$
    $\psi_y = \frac{\mathbb{I}(s_{new} < s_i):\forall s_i \in S}{|S|}$ ▷ $\mathbb{I}(\cdot)$ is an indicator function
    if $\psi_y > 1 - \alpha$ **then**
        $\mathcal{Y}_{new} \leftarrow \mathcal{Y}_{new} \cup \{y\}$
    end if
end for

return Prediction set $\mathcal{Y}_{new}$

Algorithm 4 Inductive Conformal Predictions with multiclass nonconformity score.

Input: A calibration dataset $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^{n} \subseteq \mathcal{D}_{cal}$, where $n = |\mathcal{D}_{cal}|$, $y_i \in \{1, \ldots, K\}$ (K is the number of classes) and $\boldsymbol{z}_i \in \mathbb{R}^{L \times d}$ ($L = 1$ in a single instance setup); a confidence level α, a pretrained multiclass classifier g, and a new sample $\boldsymbol{z}_{new}$.

Output: Prediction set $\mathcal{Y}_{new}$

Initialize an empty score list $S \leftarrow \emptyset$
for all samples $\langle \boldsymbol{z}_i, y_i \rangle \in \mathcal{D}_{cal}$ **do**
    $\boldsymbol{p} = g(\boldsymbol{z}_i)$ ▷ $\boldsymbol{p} \in \Delta^{K-1}$ is a vector of probabilities
    $p_z = \max_{j \neq y_i} p_j$ ▷ Maximum non-target probability, where $j \in \{1, \ldots, K\}$
    $d_p = p_{y_i} - p_z$ ▷ Probability margin
    $s_i = \frac{1 - d_p}{2}$ ▷ Non-conformity score, see Eq. 13.
    $S \leftarrow S \cup \{ s_i \}$
end for
Compute probabilities for the new samples: $\boldsymbol{p}_{new} = g(\boldsymbol{z}_{new})$
Initialize empty prediction set $\mathcal{Y}_{new} \leftarrow \emptyset$
for all $y \in \{1, \ldots, K\}$ **do**
    $p_z = \max_{j \neq y_i} p_j$ ▷ where $j \in \{1, \ldots, K\}$ and $p_j \in \boldsymbol{p}_{new}$
    $d_p = p_y - p_z$
    $s_{new} = \frac{1 - d_p}{2}$
    $\psi_y = \frac{\mathbb{I}(s_{new} < s_i) : \forall s_i \in S}{|S|}$ ▷ $\mathbb{I}(\cdot)$ is an indicator function
    if $\psi_y > 1 - \alpha$ **then**
        $\mathcal{Y}_{new} \leftarrow \mathcal{Y}_{new} \cup \{y\}$
    end if
end for
return Prediction set $\mathcal{Y}_{new}$

K More Tables on Classification Performance, Generalization Performance, and Confusion Matrices

In this section, we report detailed tables of the following results.

- Classification performance for all ⟨model, dataset⟩ pairs, including multiclass SVM (see Supplementary Tables 9– 12),
- Generalization performance of multiclass `sAwMIL` across all datasets (see Supplementary Tables 13– 15),
- Confusion matrices for all ⟨model, dataset⟩ pairs (see Supplementary Tables 16– 18).

Table 9: **Classification performance of the zero-shot prompting across datasets and models.** We report Weighted Matthew's Correlation Coefficient (W-MCC) with the 95% confidence intervals. Recall that W-MCC weighs MCC by acceptance rate (see Eq. 6). Confidence intervals are based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark W-MCC with significant confidence intervals.

Official Model Name	Type	Probe	Dataset	$CI_{.025}$	W-MCC	$CI_{.975}$
Llama-3-8B	default	Zero-shot	City Locations	0.13	**0.16**	0.19
Llama-3.2-3B	default	Zero-shot	City Locations	0.10	**0.13**	0.16
Mistral-7B-v0.3	default	Zero-shot	City Locations	0.00	0.00	0.00
Qwen-2.5-7B	default	Zero-shot	City Locations	0.59	**0.62**	0.64
Qwen-2.5-14B	default	Zero-shot	City Locations	0.73	**0.75**	0.77
Gemma-7B	default	Zero-shot	City Locations	0.00	0.00	0.00
Gemma-2-9B	default	Zero-shot	City Locations	0.50	**0.52**	0.55
Gemma-7B-it	chat	Zero-shot	City Locations	0.55	**0.58**	0.60
Gemma-2-9B-it	chat	Zero-shot	City Locations	0.74	**0.76**	0.78
Qwen-2.5-7B-Instruct	chat	Zero-shot	City Locations	0.66	**0.68**	0.70
Qwen-2.5-14B-Instruct	chat	Zero-shot	City Locations	0.81	**0.83**	0.85
Llama-3.1-8B-Instruct	chat	Zero-shot	City Locations	0.65	**0.67**	0.69
Llama-3.2-3B-Instruct	chat	Zero-shot	City Locations	0.51	**0.53**	0.56
Mistral-7B-Instruct-v0.3	chat	Zero-shot	City Locations	0.49	**0.51**	0.53
Bio-Medical-Llama-3-8B	chat	Zero-shot	City Locations	0.49	**0.51**	0.53
Llama3-Med42-8B	chat	Zero-shot	City Locations	0.58	**0.61**	0.63
Llama-3-8B	default	Zero-shot	Medical Indications	0.20	**0.23**	0.26
Llama-3.2-3B	default	Zero-shot	Medical Indications	0.09	**0.13**	0.16
Mistral-7B-v0.3	default	Zero-shot	Medical Indications	0.00	0.00	0.00
Qwen-2.5-7B	default	Zero-shot	Medical Indications	0.30	**0.33**	0.36
Qwen-2.5-14B	default	Zero-shot	Medical Indications	0.42	**0.45**	0.48
Gemma-7B	default	Zero-shot	Medical Indications	0.01	**0.02**	0.03
Gemma-2-9B	default	Zero-shot	Medical Indications	0.25	**0.28**	0.32
Gemma-7B-it	chat	Zero-shot	Medical Indications	0.26	**0.29**	0.33
Gemma-2-9B-it	chat	Zero-shot	Medical Indications	0.46	**0.50**	0.53
Qwen-2.5-7B-Instruct	chat	Zero-shot	Medical Indications	0.30	**0.33**	0.36
Qwen-2.5-14B-Instruct	chat	Zero-shot	Medical Indications	0.53	**0.56**	0.58
Llama-3.1-8B-Instruct	chat	Zero-shot	Medical Indications	0.33	**0.36**	0.40
Llama-3.2-3B-Instruct	chat	Zero-shot	Medical Indications	0.22	**0.25**	0.28
Mistral-7B-Instruct-v0.3	chat	Zero-shot	Medical Indications	0.29	**0.32**	0.34
Bio-Medical-Llama-3-8B	chat	Zero-shot	Medical Indications	0.30	**0.33**	0.36
Llama3-Med42-8B	chat	Zero-shot	Medical Indications	0.38	**0.41**	0.45
Llama-3-8B	default	Zero-shot	Word Definitions	0.03	**0.06**	0.09
Llama-3.2-3B	default	Zero-shot	Word Definitions	-0.00	0.02	0.05
Mistral-7B-v0.3	default	Zero-shot	Word Definitions	0.00	0.00	0.00
Qwen-2.5-7B	default	Zero-shot	Word Definitions	0.10	**0.13**	0.16
Qwen-2.5-14B	default	Zero-shot	Word Definitions	0.43	**0.46**	0.49
Gemma-7B	default	Zero-shot	Word Definitions	0.00	0.01	0.01
Gemma-2-9B	default	Zero-shot	Word Definitions	0.18	**0.21**	0.24
Gemma-7B-it	chat	Zero-shot	Word Definitions	0.18	**0.20**	0.23
Gemma-2-9B-it	chat	Zero-shot	Word Definitions	0.31	**0.34**	0.37
Qwen-2.5-7B-Instruct	chat	Zero-shot	Word Definitions	0.17	**0.20**	0.22
Qwen-2.5-14B-Instruct	chat	Zero-shot	Word Definitions	0.51	**0.54**	0.56
Llama-3.1-8B-Instruct	chat	Zero-shot	Word Definitions	0.17	**0.20**	0.23
Llama-3.2-3B-Instruct	chat	Zero-shot	Word Definitions	0.04	**0.07**	0.11
Mistral-7B-Instruct-v0.3	chat	Zero-shot	Word Definitions	0.22	**0.25**	0.28
Bio-Medical-Llama-3-8B	chat	Zero-shot	Word Definitions	0.11	**0.14**	0.17
Llama3-Med42-8B	chat	Zero-shot	Word Definitions	0.15	**0.18**	0.21

Table 10: **Classification performance of the multiclass `sAwMIL` probe across datasets and models.** This probe is trained and evaluated on the bag representation of the statements. We report Weighted Matthew's Correlation Coefficient (W-MCC) with the 95% confidence intervals. Recall that W-MCC weighs MCC by acceptance rate (see Eq. 6). Confidence intervals are based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark W-MCC with significant confidence intervals. 'Best Layer' column specifies the layer where a multiclass `sAwMIL` probe achieved the best W-MCC score.

Official Model Name	Type	Probe	Dataset	$CI_{.025}$	W-MCC	$CI_{.975}$	Best Layer
Llama-3-8B	default	sAwMIL	City Locations	0.87	**0.88**	0.88	16
Llama-3.2-3B	default	sAwMIL	City Locations	0.87	**0.88**	0.88	10
Mistral-7B-v0.3	default	sAwMIL	City Locations	0.87	**0.88**	0.88	17
Qwen-2.5-7B	default	sAwMIL	City Locations	0.88	**0.89**	0.89	19
Qwen-2.5-14B	default	sAwMIL	City Locations	0.86	**0.87**	0.87	28
Gemma-7B	default	sAwMIL	City Locations	0.89	**0.90**	0.90	20
Gemma-2-9B	default	sAwMIL	City Locations	0.88	**0.89**	0.89	21
Gemma-7B-it	chat	sAwMIL	City Locations	0.88	**0.89**	0.90	18
Gemma-2-9B-it	chat	sAwMIL	City Locations	0.87	**0.87**	0.88	23
Qwen-2.5-7B-Instruct	chat	sAwMIL	City Locations	0.85	**0.86**	0.87	20
Qwen-2.5-14B-Instruct	chat	sAwMIL	City Locations	0.87	**0.88**	0.88	29
Llama-3.1-8B-Instruct	chat	sAwMIL	City Locations	0.88	**0.88**	0.89	13
Llama-3.2-3B-Instruct	chat	sAwMIL	City Locations	0.86	**0.87**	0.87	12
Mistral-7B-Instruct-v0.3	chat	sAwMIL	City Locations	0.87	**0.88**	0.88	10
Bio-Medical-Llama-3-8B	chat	sAwMIL	City Locations	0.87	**0.87**	0.87	28
Llama3-Med42-8B	chat	sAwMIL	City Locations	0.88	**0.89**	0.89	14
Llama-3-8B	default	sAwMIL	Medical Indications	0.74	**0.77**	0.79	13
Llama-3.2-3B	default	sAwMIL	Medical Indications	0.65	**0.67**	0.69	10
Mistral-7B-v0.3	default	sAwMIL	Medical Indications	0.75	**0.76**	0.78	13
Qwen-2.5-7B	default	sAwMIL	Medical Indications	0.68	**0.70**	0.72	16
Qwen-2.5-14B	default	sAwMIL	Medical Indications	0.72	**0.74**	0.75	22
Gemma-7B	default	sAwMIL	Medical Indications	0.72	**0.74**	0.76	17
Gemma-2-9B	default	sAwMIL	Medical Indications	0.74	**0.76**	0.78	18
Gemma-7B-it	chat	sAwMIL	Medical Indications	0.50	**0.52**	0.54	15
Gemma-2-9B-it	chat	sAwMIL	Medical Indications	0.76	**0.78**	0.80	21
Qwen-2.5-7B-Instruct	chat	sAwMIL	Medical Indications	0.68	**0.70**	0.72	17
Qwen-2.5-14B-Instruct	chat	sAwMIL	Medical Indications	0.72	**0.74**	0.76	23
Llama-3.1-8B-Instruct	chat	sAwMIL	Medical Indications	0.78	**0.80**	0.82	18
Llama-3.2-3B-Instruct	chat	sAwMIL	Medical Indications	0.65	**0.67**	0.69	15
Mistral-7B-Instruct-v0.3	chat	sAwMIL	Medical Indications	0.75	**0.77**	0.79	16
Bio-Medical-Llama-3-8B	chat	sAwMIL	Medical Indications	0.74	**0.76**	0.78	11
Llama3-Med42-8B	chat	sAwMIL	Medical Indications	0.67	**0.69**	0.70	8
Llama-3-8B	default	sAwMIL	Word Definitions	0.83	**0.84**	0.86	13
Llama-3.2-3B	default	sAwMIL	Word Definitions	0.77	**0.79**	0.81	10
Mistral-7B-v0.3	default	sAwMIL	Word Definitions	0.84	**0.86**	0.88	13
Qwen-2.5-7B	default	sAwMIL	Word Definitions	0.84	**0.86**	0.87	16
Qwen-2.5-14B	default	sAwMIL	Word Definitions	0.84	**0.85**	0.87	21
Gemma-7B	default	sAwMIL	Word Definitions	0.77	**0.79**	0.81	14
Gemma-2-9B	default	sAwMIL	Word Definitions	0.86	**0.88**	0.89	17
Gemma-7B-it	chat	sAwMIL	Word Definitions	0.63	**0.64**	0.66	22
Gemma-2-9B-it	chat	sAwMIL	Word Definitions	0.86	**0.87**	0.89	19
Qwen-2.5-7B-Instruct	chat	sAwMIL	Word Definitions	0.84	**0.86**	0.87	18
Qwen-2.5-14B-Instruct	chat	sAwMIL	Word Definitions	0.88	**0.89**	0.90	24
Llama-3.1-8B-Instruct	chat	sAwMIL	Word Definitions	0.88	**0.89**	0.91	14
Llama-3.2-3B-Instruct	chat	sAwMIL	Word Definitions	0.85	**0.86**	0.88	12
Mistral-7B-Instruct-v0.3	chat	sAwMIL	Word Definitions	0.86	**0.87**	0.89	11
Bio-Medical-Llama-3-8B	chat	sAwMIL	Word Definitions	0.86	**0.88**	0.89	13
Llama3-Med42-8B	chat	sAwMIL	Word Definitions	0.87	**0.88**	0.90	14

Table 11: **Classification performance of the mean-difference probe with conformal prediction intervals (`MD+CP`) probe across datasets and models.** This probe is trained and evaluated on the last token's representation. We report Weighted Matthew's Correlation Coefficient (W-MCC) with the 95% confidence intervals. Recall that W-MCC weighs MCC by acceptance rate (see Eq. 6). Confidence intervals are based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark W-MCC with significant confidence intervals. 'Best Layer' column specifies the layer where a multiclass `sAwMIL` probe achieved the best W-MCC score.

Official Model Name	Type	Probe	Dataset	$CI_{.025}$	MCC	$CI_{.975}$	Best Layer
Llama-3-8B	default	MD+CP	City Locations	0.56	**0.59**	0.62	30
Llama-3.2-3B	default	MD+CP	City Locations	0.56	**0.59**	0.62	27
Mistral-7B-v0.3	default	MD+CP	City Locations	0.58	**0.61**	0.64	22
Qwen-2.5-7B	default	MD+CP	City Locations	0.50	**0.53**	0.56	21
Qwen-2.5-14B	default	MD+CP	City Locations	0.57	**0.60**	0.62	32
Gemma-7B	default	MD+CP	City Locations	0.61	**0.64**	0.66	21
Gemma-2-9B	default	MD+CP	City Locations	0.67	**0.70**	0.73	24
Gemma-7B-it	chat	MD+CP	City Locations	0.55	**0.58**	0.61	19
Gemma-2-9B-it	chat	MD+CP	City Locations	0.73	**0.75**	0.78	24
Qwen-2.5-7B-Instruct	chat	MD+CP	City Locations	0.52	**0.55**	0.58	22
Qwen-2.5-14B-Instruct	chat	MD+CP	City Locations	0.63	**0.66**	0.69	39
Llama-3.1-8B-Instruct	chat	MD+CP	City Locations	0.67	**0.69**	0.72	31
Llama-3.2-3B-Instruct	chat	MD+CP	City Locations	0.56	**0.59**	0.62	15
Mistral-7B-Instruct-v0.3	chat	MD+CP	City Locations	0.61	**0.64**	0.67	14
Bio-Medical-Llama-3-8B	chat	MD+CP	City Locations	0.65	**0.68**	0.71	22
Llama3-Med42-8B	chat	MD+CP	City Locations	0.61	**0.64**	0.67	29
Llama-3-8B	default	MD+CP	Medical Indications	0.37	**0.40**	0.43	17
Llama-3.2-3B	default	MD+CP	Medical Indications	0.34	**0.38**	0.41	14
Mistral-7B-v0.3	default	MD+CP	Medical Indications	0.37	**0.41**	0.44	20
Qwen-2.5-7B	default	MD+CP	Medical Indications	0.41	**0.44**	0.47	22
Qwen-2.5-14B	default	MD+CP	Medical Indications	0.40	**0.44**	0.47	39
Gemma-7B	default	MD+CP	Medical Indications	0.39	**0.42**	0.45	21
Gemma-2-9B	default	MD+CP	Medical Indications	0.40	**0.43**	0.46	28
Gemma-7B-it	chat	MD+CP	Medical Indications	0.27	**0.30**	0.34	17
Gemma-2-9B-it	chat	MD+CP	Medical Indications	0.40	**0.43**	0.46	21
Qwen-2.5-7B-Instruct	chat	MD+CP	Medical Indications	0.40	**0.43**	0.47	22
Qwen-2.5-14B-Instruct	chat	MD+CP	Medical Indications	0.41	**0.45**	0.48	35
Llama-3.1-8B-Instruct	chat	MD+CP	Medical Indications	0.40	**0.43**	0.47	20
Llama-3.2-3B-Instruct	chat	MD+CP	Medical Indications	0.31	**0.35**	0.38	11
Mistral-7B-Instruct-v0.3	chat	MD+CP	Medical Indications	0.35	**0.39**	0.42	15
Bio-Medical-Llama-3-8B	chat	MD+CP	Medical Indications	0.37	**0.40**	0.44	15
Llama3-Med42-8B	chat	MD+CP	Medical Indications	0.40	**0.43**	0.47	24
Llama-3-8B	default	MD+CP	Word Definitions	0.27	**0.30**	0.33	12
Llama-3.2-3B	default	MD+CP	Word Definitions	0.28	**0.31**	0.34	11
Mistral-7B-v0.3	default	MD+CP	Word Definitions	0.30	**0.33**	0.35	15
Qwen-2.5-7B	default	MD+CP	Word Definitions	0.30	**0.33**	0.36	20
Qwen-2.5-14B	default	MD+CP	Word Definitions	0.32	**0.35**	0.38	28
Gemma-7B	default	MD+CP	Word Definitions	0.31	**0.34**	0.37	15
Gemma-2-9B	default	MD+CP	Word Definitions	0.31	**0.34**	0.37	23
Gemma-7B-it	chat	MD+CP	Word Definitions	0.32	**0.35**	0.37	17
Gemma-2-9B-it	chat	MD+CP	Word Definitions	0.32	**0.35**	0.37	19
Qwen-2.5-7B-Instruct	chat	MD+CP	Word Definitions	0.33	**0.35**	0.38	21
Qwen-2.5-14B-Instruct	chat	MD+CP	Word Definitions	0.32	**0.35**	0.37	31
Llama-3.1-8B-Instruct	chat	MD+CP	Word Definitions	0.33	**0.36**	0.39	13
Llama-3.2-3B-Instruct	chat	MD+CP	Word Definitions	0.17	**0.20**	0.24	10
Mistral-7B-Instruct-v0.3	chat	MD+CP	Word Definitions	0.31	**0.34**	0.36	18
Bio-Medical-Llama-3-8B	chat	MD+CP	Word Definitions	0.27	**0.30**	0.33	19
Llama3-Med42-8B	chat	MD+CP	Word Definitions	0.36	**0.38**	0.41	24

Table 12: **Classification performance of the multiclass SVM probe across datasets and models.** This probe is trained on the last token's representations and evaluated on the bag representation of the statements. We report Weighted Matthew's Correlation Coefficient (W-MCC) with the 95% confidence intervals. Recall that W-MCC weighs MCC by acceptance rate (see Eq. 6). Confidence intervals are based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark W-MCC with significant confidence intervals. 'Best Layer' column specifies the layer where a multiclass `sAwMIL` probe achieved the best W-MCC score.

Official Model Name	Type	Probe	Dataset	$CI_{.025}$	W-MCC	$CI_{.975}$	Best Layer
Llama-3-8B	default	SVM	City Locations	0.79	**0.81**	0.83	13
Llama-3.2-3B	default	SVM	City Locations	0.70	**0.73**	0.75	27
Mistral-7B-v0.3	default	SVM	City Locations	0.84	**0.86**	0.87	13
Qwen-2.5-7B	default	SVM	City Locations	0.66	**0.68**	0.71	26
Qwen-2.5-14B	default	SVM	City Locations	0.78	**0.80**	0.82	29
Gemma-7B	default	SVM	City Locations	0.69	**0.72**	0.74	21
Gemma-2-9B	default	SVM	City Locations	0.78	**0.81**	0.83	14
Gemma-7B-it	chat	SVM	City Locations	0.47	**0.49**	0.51	14
Gemma-2-9B-it	chat	SVM	City Locations	0.77	**0.79**	0.81	27
Qwen-2.5-7B-Instruct	chat	SVM	City Locations	0.00	0.00	0.00	2
Qwen-2.5-14B-Instruct	chat	SVM	City Locations	0.65	**0.67**	0.69	31
Llama-3.1-8B-Instruct	chat	SVM	City Locations	0.58	**0.60**	0.61	15
Llama-3.2-3B-Instruct	chat	SVM	City Locations	0.58	**0.60**	0.62	19
Mistral-7B-Instruct-v0.3	chat	SVM	City Locations	0.87	**0.88**	0.89	17
Bio-Medical-Llama-3-8B	chat	SVM	City Locations	0.46	**0.49**	0.51	12
Llama3-Med42-8B	chat	SVM	City Locations	0.72	**0.74**	0.76	31
Llama-3-8B	default	SVM	Medical Indications	0.53	**0.56**	0.59	13
Llama-3.2-3B	default	SVM	Medical Indications	0.53	**0.55**	0.57	11
Mistral-7B-v0.3	default	SVM	Medical Indications	0.63	**0.66**	0.68	11
Qwen-2.5-7B	default	SVM	Medical Indications	0.54	**0.56**	0.59	17
Qwen-2.5-14B	default	SVM	Medical Indications	0.62	**0.65**	0.67	24
Gemma-7B	default	SVM	Medical Indications	0.66	**0.69**	0.72	15
Gemma-2-9B	default	SVM	Medical Indications	0.62	**0.65**	0.68	21
Gemma-7B-it	chat	SVM	Medical Indications	0.37	**0.41**	0.45	15
Gemma-2-9B-it	chat	SVM	Medical Indications	0.69	**0.71**	0.74	23
Qwen-2.5-7B-Instruct	chat	SVM	Medical Indications	0.16	**0.19**	0.22	19
Qwen-2.5-14B-Instruct	chat	SVM	Medical Indications	0.32	**0.35**	0.38	11
Llama-3.1-8B-Instruct	chat	SVM	Medical Indications	0.50	**0.52**	0.54	12
Llama-3.2-3B-Instruct	chat	SVM	Medical Indications	0.43	**0.45**	0.48	20
Mistral-7B-Instruct-v0.3	chat	SVM	Medical Indications	0.70	**0.72**	0.75	18
Bio-Medical-Llama-3-8B	chat	SVM	Medical Indications	0.50	**0.52**	0.54	16
Llama3-Med42-8B	chat	SVM	Medical Indications	0.48	**0.51**	0.53	16
Llama-3-8B	default	SVM	Word Definitions	0.76	**0.79**	0.81	15
Llama-3.2-3B	default	SVM	Word Definitions	0.72	**0.74**	0.76	13
Mistral-7B-v0.3	default	SVM	Word Definitions	0.74	**0.76**	0.78	11
Qwen-2.5-7B	default	SVM	Word Definitions	0.72	**0.74**	0.77	16
Qwen-2.5-14B	default	SVM	Word Definitions	0.78	**0.81**	0.83	22
Gemma-7B	default	SVM	Word Definitions	0.74	**0.76**	0.79	16
Gemma-2-9B	default	SVM	Word Definitions	0.79	**0.81**	0.83	23
Gemma-7B-it	chat	SVM	Word Definitions	0.63	**0.65**	0.68	25
Gemma-2-9B-it	chat	SVM	Word Definitions	0.63	**0.65**	0.67	41
Qwen-2.5-7B-Instruct	chat	SVM	Word Definitions	0.18	**0.21**	0.23	6
Qwen-2.5-14B-Instruct	chat	SVM	Word Definitions	0.52	**0.55**	0.57	25
Llama-3.1-8B-Instruct	chat	SVM	Word Definitions	0.00	0.04	0.06	29
Llama-3.2-3B-Instruct	chat	SVM	Word Definitions	0.59	**0.62**	0.64	11
Mistral-7B-Instruct-v0.3	chat	SVM	Word Definitions	0.78	**0.80**	0.81	18
Bio-Medical-Llama-3-8B	chat	SVM	Word Definitions	0.58	**0.60**	0.62	5
Llama3-Med42-8B	chat	SVM	Word Definitions	0.71	**0.73**	0.75	28

Table 13: **Generalization performance of the multiclass `sAwMIL` trained on the *City Locations* dataset**. The performance is measured by the Matthew's Correlation Coefficient (MCC) with 95% confidence intervals, based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark MCC with significant confidence intervals. The 'Rel. Depth' column specifies the relative depth of the layer where the multiclass `sAwMIL` probe achieves the best MCC score.

Model Name	Training Dataset	Test Dataset	$CI_{.025}$	MCC	$CI_{.975}$	Rel. Depth
Gemma-7B-it	City Locations	City Locations	0.93	**0.94**	0.95	0.59
Gemma-7B-it	City Locations	Medical Indications	0.43	**0.46**	0.49	0.63
Gemma-7B-it	City Locations	Word Definitions	0.44	**0.47**	0.50	0.63
Gemma-2-9B-it	City Locations	City Locations	0.96	**0.97**	0.98	0.56
Gemma-2-9B-it	City Locations	Medical Indications	0.67	**0.70**	0.73	0.44
Gemma-2-9B-it	City Locations	Word Definitions	0.69	**0.71**	0.74	0.49
Llama-3.2-3B-Instruct	City Locations	City Locations	0.95	**0.96**	0.97	0.52
Llama-3.2-3B-Instruct	City Locations	Medical Indications	0.59	**0.62**	0.65	0.56
Llama-3.2-3B-Instruct	City Locations	Word Definitions	0.62	**0.65**	0.67	0.48
Llama3-Med42-8B	City Locations	City Locations	0.96	**0.97**	0.98	0.42
Llama3-Med42-8B	City Locations	Medical Indications	0.75	**0.78**	0.81	0.90
Llama3-Med42-8B	City Locations	Word Definitions	0.75	**0.78**	0.80	0.45
Llama-3.1-8B-Instruct	City Locations	City Locations	0.96	**0.97**	0.98	0.48
Llama-3.1-8B-Instruct	City Locations	Medical Indications	0.73	**0.75**	0.78	0.52
Llama-3.1-8B-Instruct	City Locations	Word Definitions	0.79	**0.81**	0.83	0.42
Bio-Medical-Llama-3-8B	City Locations	City Locations	0.96	**0.97**	0.98	0.97
Bio-Medical-Llama-3-8B	City Locations	Medical Indications	0.59	**0.62**	0.65	0.42
Bio-Medical-Llama-3-8B	City Locations	Word Definitions	0.53	**0.56**	0.59	0.26
Mistral-7B-Instruct-v0.3	City Locations	City Locations	0.95	**0.96**	0.97	0.48
Mistral-7B-Instruct-v0.3	City Locations	Medical Indications	0.72	**0.75**	0.78	0.55
Mistral-7B-Instruct-v0.3	City Locations	Word Definitions	0.69	**0.71**	0.74	0.35
Qwen-2.5-7B-Instruct	City Locations	City Locations	0.94	**0.95**	0.96	0.67
Qwen-2.5-7B-Instruct	City Locations	Medical Indications	0.71	**0.74**	0.76	0.70
Qwen-2.5-7B-Instruct	City Locations	Word Definitions	0.62	**0.64**	0.67	0.70
Qwen-2.5-14B-Instruct	City Locations	City Locations	0.96	**0.97**	0.98	0.62
Qwen-2.5-14B-Instruct	City Locations	Medical Indications	0.74	**0.77**	0.80	0.64
Qwen-2.5-14B-Instruct	City Locations	Word Definitions	0.73	**0.75**	0.77	0.60
Gemma-7B	City Locations	City Locations	0.96	**0.97**	0.98	0.74
Gemma-7B	City Locations	Medical Indications	0.55	**0.58**	0.60	0.41
Gemma-7B	City Locations	Word Definitions	0.61	**0.63**	0.66	0.52
Gemma-2-9B	City Locations	City Locations	0.97	**0.98**	0.99	0.63
Gemma-2-9B	City Locations	Medical Indications	0.52	**0.55**	0.58	0.44
Gemma-2-9B	City Locations	Word Definitions	0.54	**0.56**	0.58	0.27
Llama-3.2-3B	City Locations	City Locations	0.95	**0.96**	0.97	0.37
Llama-3.2-3B	City Locations	Medical Indications	0.35	**0.38**	0.41	0.33
Llama-3.2-3B	City Locations	Word Definitions	0.48	**0.50**	0.52	0.30
Llama-3-8B	City Locations	City Locations	0.96	**0.97**	0.98	0.32
Llama-3-8B	City Locations	Medical Indications	0.59	**0.62**	0.65	0.35
Llama-3-8B	City Locations	Word Definitions	0.56	**0.58**	0.61	0.26
Mistral-7B-v0.3	City Locations	City Locations	0.96	**0.97**	0.98	0.42
Mistral-7B-v0.3	City Locations	Medical Indications	0.50	**0.53**	0.55	0.39
Mistral-7B-v0.3	City Locations	Word Definitions	0.58	**0.61**	0.63	0.39
Qwen-2.5-7B	City Locations	City Locations	0.94	**0.95**	0.96	0.70
Qwen-2.5-7B	City Locations	Medical Indications	0.48	**0.52**	0.55	0.59
Qwen-2.5-7B	City Locations	Word Definitions	0.51	**0.54**	0.57	0.59
Qwen-2.5-14B	City Locations	City Locations	0.95	**0.96**	0.97	0.79
Qwen-2.5-14B	City Locations	Medical Indications	0.59	**0.62**	0.65	0.45
Qwen-2.5-14B	City Locations	Word Definitions	0.62	**0.64**	0.67	0.43

Table 14: **Generalization performance of the multiclass `sAwMIL` trained on the *Medical Indications* dataset.** The performance is measured by the Matthew's Correlation Coefficient (MCC) with 95% confidence intervals, based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark MCC significant confidence intervals. The 'Rel. Depth' column specifies the relative depth of the layer where a multiclass `sAwMIL` probe achieves the best MCC score.

Model Name	Training Dataset	Test Dataset	$CI_{.025}$	MCC	$CI_{.975}$	Rel. Depth
Gemma-7B-it	Medical Indications	City Locations	0.90	**0.91**	0.93	0.70
Gemma-7B-it	Medical Indications	Medical Indications	0.65	**0.68**	0.72	0.59
Gemma-7B-it	Medical Indications	Word Definitions	0.64	**0.67**	0.69	0.59
Gemma-2-9B-it	Medical Indications	City Locations	0.91	**0.92**	0.94	0.63
Gemma-2-9B-it	Medical Indications	Medical Indications	0.81	**0.83**	0.85	0.49
Gemma-2-9B-it	Medical Indications	Word Definitions	0.78	**0.80**	0.82	0.61
Llama-3.2-3B-Instruct	Medical Indications	City Locations	0.80	**0.82**	0.85	0.41
Llama-3.2-3B-Instruct	Medical Indications	Medical Indications	0.73	**0.76**	0.79	0.48
Llama-3.2-3B-Instruct	Medical Indications	Word Definitions	0.65	**0.67**	0.70	0.48
Llama3-Med42-8B	Medical Indications	City Locations	0.95	**0.96**	0.97	0.52
Llama3-Med42-8B	Medical Indications	Medical Indications	0.80	**0.83**	0.85	0.45
Llama3-Med42-8B	Medical Indications	Word Definitions	0.76	**0.78**	0.80	0.45
Llama-3.1-8B-Instruct	Medical Indications	City Locations	0.92	**0.93**	0.94	0.55
Llama-3.1-8B-Instruct	Medical Indications	Medical Indications	0.81	**0.83**	0.85	0.55
Llama-3.1-8B-Instruct	Medical Indications	Word Definitions	0.77	**0.79**	0.81	0.42
Bio-Medical-Llama-3-8B	Medical Indications	City Locations	0.85	**0.86**	0.88	0.39
Bio-Medical-Llama-3-8B	Medical Indications	Medical Indications	0.78	**0.81**	0.83	0.81
Bio-Medical-Llama-3-8B	Medical Indications	Word Definitions	0.70	**0.73**	0.75	0.32
Mistral-7B-Instruct-v0.3	Medical Indications	City Locations	0.93	**0.94**	0.95	0.39
Mistral-7B-Instruct-v0.3	Medical Indications	Medical Indications	0.78	**0.80**	0.83	0.45
Mistral-7B-Instruct-v0.3	Medical Indications	Word Definitions	0.74	**0.77**	0.79	0.45
Qwen-2.5-7B-Instruct	Medical Indications	City Locations	0.78	**0.80**	0.83	0.52
Qwen-2.5-7B-Instruct	Medical Indications	Medical Indications	0.74	**0.77**	0.79	0.67
Qwen-2.5-7B-Instruct	Medical Indications	Word Definitions	0.64	**0.67**	0.69	0.63
Qwen-2.5-14B-Instruct	Medical Indications	City Locations	0.90	**0.92**	0.93	0.49
Qwen-2.5-14B-Instruct	Medical Indications	Medical Indications	0.79	**0.82**	0.84	0.57
Qwen-2.5-14B-Instruct	Medical Indications	Word Definitions	0.76	**0.79**	0.81	0.57
Gemma-7B	Medical Indications	City Locations	0.60	**0.63**	0.66	0.56
Gemma-7B	Medical Indications	Medical Indications	0.75	**0.78**	0.80	0.63
Gemma-7B	Medical Indications	Word Definitions	0.55	**0.57**	0.59	0.70
Gemma-2-9B	Medical Indications	City Locations	0.83	**0.85**	0.87	0.56
Gemma-2-9B	Medical Indications	Medical Indications	0.77	**0.80**	0.82	0.44
Gemma-2-9B	Medical Indications	Word Definitions	0.64	**0.67**	0.69	0.39
Llama-3.2-3B	Medical Indications	City Locations	0.49	**0.51**	0.53	0.41
Llama-3.2-3B	Medical Indications	Medical Indications	0.74	**0.76**	0.79	0.44
Llama-3.2-3B	Medical Indications	Word Definitions	0.59	**0.62**	0.64	0.52
Llama-3-8B	Medical Indications	City Locations	0.75	**0.77**	0.80	0.45
Llama-3-8B	Medical Indications	Medical Indications	0.77	**0.80**	0.83	0.39
Llama-3-8B	Medical Indications	Word Definitions	0.63	**0.65**	0.68	0.26
Mistral-7B-v0.3	Medical Indications	City Locations	0.58	**0.60**	0.63	0.42
Mistral-7B-v0.3	Medical Indications	Medical Indications	0.77	**0.80**	0.82	0.42
Mistral-7B-v0.3	Medical Indications	Word Definitions	0.63	**0.65**	0.68	0.42
Qwen-2.5-7B	Medical Indications	City Locations	0.80	**0.82**	0.84	0.67
Qwen-2.5-7B	Medical Indications	Medical Indications	0.76	**0.78**	0.81	0.63
Qwen-2.5-7B	Medical Indications	Word Definitions	0.59	**0.62**	0.65	0.74
Qwen-2.5-14B	Medical Indications	City Locations	0.80	**0.82**	0.84	0.70
Qwen-2.5-14B	Medical Indications	Medical Indications	0.76	**0.79**	0.82	0.60
Qwen-2.5-14B	Medical Indications	Word Definitions	0.70	**0.72**	0.75	0.60

Table 15: **Generalization performance of the multiclass `sAwMIL` trained on the *Word Definitions* dataset.** The performance is measured by the Matthew's Correlation Coefficient (MCC) with 95% confidence intervals, based on bootstrapping with $n = 1{,}000$ samples. The **bold** values mark MCC with significant confidence intervals. The 'Rel. Depth' column specifies the relative depth of the layer where a multiclass `sAwMIL` probe achieves the best MCC score.

Model Name	Training Dataset	Test Dataset	$CI_{.025}$	MCC	$CI_{.975}$	Rel. Depth
Gemma-7B-it	Word Definitions	City Locations	0.90	**0.92**	0.93	0.70
Gemma-7B-it	Word Definitions	Medical Indications	0.53	**0.56**	0.60	0.67
Gemma-7B-it	Word Definitions	Word Definitions	0.78	**0.80**	0.82	0.67
Gemma-2-9B-it	Word Definitions	City Locations	0.94	**0.96**	0.97	0.56
Gemma-2-9B-it	Word Definitions	Medical Indications	0.65	**0.69**	0.71	0.41
Gemma-2-9B-it	Word Definitions	Word Definitions	0.88	**0.90**	0.91	0.54
Llama-3.2-3B-Instruct	Word Definitions	City Locations	0.84	**0.86**	0.88	0.48
Llama-3.2-3B-Instruct	Word Definitions	Medical Indications	0.60	**0.63**	0.66	0.44
Llama-3.2-3B-Instruct	Word Definitions	Word Definitions	0.85	**0.86**	0.88	0.44
Llama3-Med42-8B	Word Definitions	City Locations	0.94	**0.95**	0.97	0.35
Llama3-Med42-8B	Word Definitions	Medical Indications	0.76	**0.79**	0.81	0.45
Llama3-Med42-8B	Word Definitions	Word Definitions	0.87	**0.89**	0.91	0.45
Llama-3.1-8B-Instruct	Word Definitions	City Locations	0.95	**0.96**	0.97	0.45
Llama-3.1-8B-Instruct	Word Definitions	Medical Indications	0.72	**0.75**	0.77	0.32
Llama-3.1-8B-Instruct	Word Definitions	Word Definitions	0.90	**0.91**	0.93	0.45
Bio-Medical-Llama-3-8B	Word Definitions	City Locations	0.78	**0.81**	0.83	0.35
Bio-Medical-Llama-3-8B	Word Definitions	Medical Indications	0.70	**0.73**	0.76	0.32
Bio-Medical-Llama-3-8B	Word Definitions	Word Definitions	0.87	**0.89**	0.90	0.39
Mistral-7B-Instruct-v0.3	Word Definitions	City Locations	0.93	**0.94**	0.96	0.48
Mistral-7B-Instruct-v0.3	Word Definitions	Medical Indications	0.74	**0.77**	0.80	0.52
Mistral-7B-Instruct-v0.3	Word Definitions	Word Definitions	0.87	**0.88**	0.90	0.35
Qwen-2.5-7B-Instruct	Word Definitions	City Locations	0.85	**0.87**	0.88	0.63
Qwen-2.5-7B-Instruct	Word Definitions	Medical Indications	0.74	**0.76**	0.79	0.67
Qwen-2.5-7B-Instruct	Word Definitions	Word Definitions	0.85	**0.87**	0.88	0.63
Qwen-2.5-14B-Instruct	Word Definitions	City Locations	0.95	**0.96**	0.97	0.66
Qwen-2.5-14B-Instruct	Word Definitions	Medical Indications	0.77	**0.79**	0.82	0.66
Qwen-2.5-14B-Instruct	Word Definitions	Word Definitions	0.88	**0.90**	0.92	0.51
Gemma-7B	Word Definitions	City Locations	0.88	**0.90**	0.91	0.56
Gemma-7B	Word Definitions	Medical Indications	0.59	**0.63**	0.66	0.48
Gemma-7B	Word Definitions	Word Definitions	0.81	**0.83**	0.85	0.56
Gemma-2-9B	Word Definitions	City Locations	0.92	**0.93**	0.94	0.56
Gemma-2-9B	Word Definitions	Medical Indications	0.70	**0.73**	0.75	0.46
Gemma-2-9B	Word Definitions	Word Definitions	0.86	**0.88**	0.90	0.41
Llama-3.2-3B	Word Definitions	City Locations	0.76	**0.78**	0.81	0.41
Llama-3.2-3B	Word Definitions	Medical Indications	0.70	**0.73**	0.76	0.41
Llama-3.2-3B	Word Definitions	Word Definitions	0.79	**0.81**	0.83	0.41
Llama-3-8B	Word Definitions	City Locations	0.84	**0.86**	0.88	0.39
Llama-3-8B	Word Definitions	Medical Indications	0.74	**0.76**	0.79	0.39
Llama-3-8B	Word Definitions	Word Definitions	0.85	**0.87**	0.89	0.39
Mistral-7B-v0.3	Word Definitions	City Locations	0.77	**0.79**	0.81	0.52
Mistral-7B-v0.3	Word Definitions	Medical Indications	0.73	**0.76**	0.78	0.39
Mistral-7B-v0.3	Word Definitions	Word Definitions	0.85	**0.87**	0.89	0.45
Qwen-2.5-7B	Word Definitions	City Locations	0.90	**0.92**	0.93	0.67
Qwen-2.5-7B	Word Definitions	Medical Indications	0.72	**0.74**	0.77	0.59
Qwen-2.5-7B	Word Definitions	Word Definitions	0.85	**0.87**	0.88	0.59
Qwen-2.5-14B	Word Definitions	City Locations	0.92	**0.94**	0.95	0.45
Qwen-2.5-14B	Word Definitions	Medical Indications	0.72	**0.75**	0.78	0.66
Qwen-2.5-14B	Word Definitions	Word Definitions	0.85	**0.87**	0.88	0.47

Table 16: **Row-wise confusion matrices for zero-shot prompting across all ⟨model, dataset⟩ pairs.** Each row corresponds to a specific model and a dataset. Columns are grouped by the ground-truth labels (*True*, *False*, *Neither*) with groups of subcolumns that specify the distribution of predictions (*true*, *false*, *neither*, *abstain*). For each statement in a dataset, the predicted class is the class with the highest probability (as estimated by zero-shot prompting). The values in each group of four subcolumns sum to 1 because they are normalized counts. For example, in the first row under the *True* ground-truth label, we see that *true* predictions have the value of 0.89 – that means that 89% of all the true statements are classified as true.

	Ground-truth label →	True				False				Neither			
	Predicted →	True	False	Neither	Abstain	True	False	Neither	Abstain	True	False	Neither	Abstain
Model ↓	Data Set ↓												
Bio-Medical-Llama-3-8B	City Locations	0.89	0.11	0.00	0.00	0.06	0.94	0.00	0.00	0.18	0.82	0.00	0.00
	Medical Indications	0.79	0.21	0.00	0.00	0.32	0.68	0.00	0.00	0.27	0.73	0.00	0.00
	Word Definitions	0.61	0.39	0.00	0.00	0.37	0.63	0.00	0.00	0.05	0.95	0.00	0.00
Gemma-2-9B	City Locations	0.50	0.10	0.39	0.01	0.03	0.53	0.43	0.01	0.05	0.00	0.93	0.02
	Medical Indications	0.70	0.12	0.18	0.00	0.34	0.51	0.14	0.00	0.57	0.19	0.24	0.00
	Word Definitions	0.36	0.20	0.40	0.05	0.17	0.26	0.53	0.04	0.14	0.12	0.72	0.01
Gemma-2-9B-it	City Locations	0.98	0.02	0.00	0.00	0.03	0.97	0.00	0.00	0.06	0.45	0.49	0.00
	Medical Indications	0.87	0.12	0.01	0.00	0.25	0.75	0.00	0.00	0.23	0.59	0.18	0.00
	Word Definitions	0.76	0.16	0.09	0.00	0.25	0.70	0.06	0.00	0.10	0.65	0.25	0.00
Gemma-7B	City Locations	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00
	Medical Indications	0.16	0.00	0.00	0.84	0.09	0.00	0.00	0.91	0.03	0.00	0.00	0.96
	Word Definitions	0.03	0.00	0.00	0.97	0.03	0.01	0.00	0.96	0.00	0.00	0.00	1.00
Gemma-7B-it	City Locations	0.76	0.23	0.01	0.00	0.05	0.95	0.00	0.00	0.04	0.64	0.32	0.00
	Medical Indications	0.69	0.30	0.01	0.00	0.27	0.73	0.00	0.00	0.37	0.61	0.02	0.00
	Word Definitions	0.27	0.63	0.09	0.01	0.09	0.89	0.02	0.00	0.04	0.77	0.15	0.04
Llama-3-8B	City Locations	0.35	0.65	0.00	0.00	0.22	0.78	0.00	0.00	0.47	0.52	0.00	0.01
	Medical Indications	0.33	0.67	0.00	0.00	0.08	0.92	0.00	0.00	0.19	0.81	0.00	0.00
	Word Definitions	0.45	0.55	0.00	0.00	0.33	0.67	0.00	0.00	0.37	0.63	0.00	0.00
Llama-3.1-8B-Instruct	City Locations	0.95	0.05	0.00	0.00	0.03	0.97	0.00	0.00	0.08	0.65	0.27	0.00
	Medical Indications	0.54	0.46	0.00	0.00	0.07	0.93	0.00	0.00	0.13	0.86	0.01	0.00
	Word Definitions	0.54	0.46	0.00	0.00	0.21	0.79	0.00	0.00	0.06	0.94	0.00	0.00
Llama-3.2-3B	City Locations	0.29	0.71	0.00	0.00	0.11	0.89	0.00	0.00	0.43	0.57	0.00	0.00
	Medical Indications	0.46	0.54	0.00	0.00	0.34	0.66	0.00	0.00	0.50	0.50	0.00	0.00
	Word Definitions	0.48	0.52	0.00	0.00	0.44	0.56	0.00	0.00	0.46	0.54	0.00	0.00
Llama-3.2-3B-Instruct	City Locations	0.93	0.07	0.00	0.00	0.15	0.85	0.00	0.00	0.04	0.84	0.13	0.00
	Medical Indications	0.35	0.65	0.00	0.00	0.07	0.93	0.00	0.00	0.15	0.85	0.00	0.00
	Word Definitions	0.60	0.40	0.00	0.00	0.46	0.54	0.00	0.00	0.06	0.93	0.01	0.00
Llama3-Med42-8B	City Locations	0.96	0.04	0.00	0.00	0.04	0.96	0.00	0.00	0.22	0.64	0.14	0.00
	Medical Indications	0.69	0.31	0.00	0.00	0.13	0.87	0.00	0.00	0.17	0.82	0.00	0.00
	Word Definitions	0.47	0.52	0.01	0.00	0.18	0.81	0.01	0.00	0.08	0.89	0.03	0.00
Mistral-7B-Instruct-v0.3	City Locations	0.88	0.00	0.09	0.03	0.04	0.09	0.50	0.37	0.08	0.00	0.90	0.02
	Medical Indications	0.47	0.04	0.49	0.00	0.07	0.09	0.85	0.00	0.03	0.00	0.97	0.00
	Word Definitions	0.63	0.02	0.33	0.02	0.40	0.01	0.55	0.04	0.25	0.00	0.73	0.02
Mistral-7B-v0.3	City Locations	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00
	Medical Indications	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00
	Word Definitions	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00
Qwen-2.5-14B	City Locations	0.95	0.05	0.00	0.00	0.02	0.98	0.00	0.00	0.04	0.47	0.49	0.00
	Medical Indications	0.53	0.46	0.01	0.00	0.04	0.95	0.00	0.00	0.00	0.77	0.23	0.00
	Word Definitions	0.50	0.41	0.09	0.00	0.05	0.90	0.05	0.00	0.00	0.53	0.47	0.00
Qwen-2.5-14B-Instruct	City Locations	0.93	0.07	0.00	0.00	0.02	0.98	0.00	0.00	0.00	0.28	0.71	0.00
	Medical Indications	0.63	0.23	0.14	0.00	0.04	0.87	0.09	0.00	0.00	0.46	0.54	0.00
	Word Definitions	0.55	0.32	0.13	0.00	0.06	0.89	0.06	0.00	0.01	0.37	0.62	0.00
Qwen-2.5-7B	City Locations	0.92	0.07	0.01	0.00	0.03	0.96	0.01	0.00	0.09	0.68	0.23	0.00
	Medical Indications	0.56	0.44	0.00	0.00	0.11	0.89	0.00	0.00	0.25	0.75	0.00	0.00
	Word Definitions	0.60	0.39	0.01	0.00	0.31	0.67	0.02	0.00	0.35	0.64	0.01	0.00
Qwen-2.5-7B-Instruct	City Locations	0.89	0.11	0.00	0.00	0.02	0.98	0.00	0.00	0.02	0.57	0.37	0.03
	Medical Indications	0.41	0.56	0.02	0.00	0.04	0.94	0.02	0.00	0.01	0.92	0.06	0.01
	Word Definitions	0.56	0.39	0.04	0.01	0.27	0.70	0.03	0.01	0.25	0.61	0.12	0.03

Table 17: **Row-wise confusion matrices for mean-difference probe with conformal prediction intervals (MD+CP) across all ⟨model-dataset pairs⟩.** Each row corresponds to a specific model and a dataset. Columns are grouped by the ground-truth labels (*True*, *False*, *Neither*) with groups of subcolumns that specify the distribution of predictions (*true*, *false*, *neither*, *abstain*). For each statement in a dataset, the predicted class is the class with the highest probability (as estimated by MD+CP). The values in each group of four subcolumns sum to 1 because they are normalized counts. For example, in the first row under the *True* ground-truth label, we see that *true* predictions have the value of 0.89 – that means that 89% of all the true statements are classified as true.

	True Labels →	True				False				Neither			
	Predicted →	True	False	Neither	Abstain	True	False	Neither	Abstain	True	False	Neither	Abstain
Model ↓	Data Set ↓												
Bio-Medical-Llama-3-8B	City Locations	0.89	0.01	0.10	0.00	0.01	0.91	0.08	0.00	0.17	0.39	0.44	0.00
	Medical Indications	0.69	0.13	0.18	0.00	0.09	0.76	0.16	0.00	0.22	0.59	0.20	0.00
	Word Definitions	0.73	0.10	0.17	0.00	0.08	0.73	0.19	0.00	0.37	0.42	0.21	0.00
Gemma-2-9B	City Locations	0.91	0.00	0.09	0.00	0.00	0.91	0.09	0.00	0.19	0.25	0.56	0.00
	Medical Indications	0.76	0.09	0.15	0.00	0.10	0.75	0.15	0.00	0.11	0.77	0.12	0.00
	Word Definitions	0.79	0.10	0.11	0.00	0.09	0.81	0.10	0.00	0.46	0.39	0.15	0.00
Gemma-2-9B-it	City Locations	0.91	0.01	0.08	0.00	0.01	0.91	0.08	0.00	0.06	0.59	0.35	0.00
	Medical Indications	0.70	0.09	0.21	0.00	0.09	0.72	0.19	0.00	0.42	0.29	0.29	0.00
	Word Definitions	0.76	0.10	0.14	0.00	0.10	0.75	0.15	0.00	0.23	0.61	0.16	0.00
Gemma-7B	City Locations	0.92	0.01	0.08	0.00	0.01	0.91	0.08	0.00	0.13	0.55	0.33	0.00
	Medical Indications	0.71	0.10	0.19	0.00	0.08	0.71	0.20	0.00	0.19	0.52	0.29	0.00
	Word Definitions	0.70	0.10	0.20	0.00	0.08	0.75	0.17	0.00	0.36	0.36	0.28	0.00
Gemma-7B-it	City Locations	0.91	0.01	0.08	0.00	0.02	0.88	0.10	0.00	0.31	0.42	0.27	0.00
	Medical Indications	0.58	0.12	0.30	0.00	0.11	0.59	0.30	0.00	0.32	0.35	0.32	0.00
	Word Definitions	0.76	0.12	0.12	0.00	0.11	0.79	0.10	0.00	0.31	0.48	0.21	0.00
Llama-3-8B	City Locations	0.90	0.02	0.07	0.00	0.00	0.93	0.07	0.00	0.65	0.22	0.14	0.00
	Medical Indications	0.73	0.10	0.17	0.00	0.11	0.74	0.15	0.00	0.64	0.26	0.11	0.00
	Word Definitions	0.73	0.09	0.18	0.00	0.07	0.74	0.19	0.00	0.47	0.31	0.22	0.00
Llama-3.1-8B-Instruct	City Locations	0.90	0.01	0.09	0.00	0.01	0.91	0.08	0.00	0.38	0.16	0.46	0.00
	Medical Indications	0.76	0.11	0.12	0.00	0.10	0.78	0.12	0.00	0.49	0.39	0.12	0.00
	Word Definitions	0.81	0.07	0.11	0.00	0.08	0.80	0.12	0.00	0.24	0.59	0.17	0.00
Llama-3.2-3B	City Locations	0.91	0.02	0.08	0.00	0.02	0.89	0.08	0.00	0.12	0.71	0.17	0.00
	Medical Indications	0.72	0.12	0.16	0.00	0.12	0.71	0.17	0.00	0.19	0.64	0.17	0.00
	Word Definitions	0.73	0.09	0.18	0.00	0.11	0.71	0.18	0.00	0.31	0.44	0.26	0.00
Llama-3.2-3B-Instruct	City Locations	0.92	0.02	0.06	0.00	0.04	0.89	0.07	0.00	0.45	0.34	0.21	0.00
	Medical Indications	0.59	0.11	0.31	0.00	0.11	0.62	0.27	0.00	0.32	0.38	0.30	0.00
	Word Definitions	0.60	0.10	0.30	0.00	0.09	0.63	0.28	0.00	0.37	0.40	0.23	0.00
Llama3-Med42-8B	City Locations	0.89	0.01	0.09	0.00	0.01	0.92	0.07	0.00	0.16	0.53	0.31	0.00
	Medical Indications	0.75	0.12	0.13	0.00	0.10	0.78	0.12	0.00	0.11	0.83	0.05	0.00
	Word Definitions	0.84	0.09	0.07	0.00	0.08	0.87	0.05	0.00	0.59	0.32	0.09	0.00
Mistral-7B-Instruct-v0.3	City Locations	0.92	0.01	0.07	0.00	0.02	0.90	0.08	0.00	0.67	0.14	0.19	0.00
	Medical Indications	0.68	0.10	0.22	0.00	0.11	0.70	0.19	0.00	0.31	0.52	0.17	0.00
	Word Definitions	0.78	0.11	0.11	0.00	0.11	0.79	0.11	0.00	0.33	0.50	0.16	0.00
Mistral-7B-v0.3	City Locations	0.91	0.01	0.08	0.00	0.02	0.92	0.06	0.00	0.63	0.20	0.17	0.00
	Medical Indications	0.70	0.12	0.18	0.00	0.10	0.73	0.17	0.00	0.23	0.54	0.22	0.00
	Word Definitions	0.72	0.08	0.20	0.00	0.08	0.77	0.15	0.00	0.42	0.34	0.24	0.00
Qwen-2.5-14B	City Locations	0.91	0.02	0.07	0.00	0.02	0.89	0.09	0.00	0.48	0.26	0.27	0.00
	Medical Indications	0.74	0.09	0.17	0.00	0.10	0.73	0.17	0.00	0.37	0.42	0.20	0.00
	Word Definitions	0.81	0.09	0.10	0.00	0.11	0.80	0.09	0.00	0.40	0.47	0.13	0.00
Qwen-2.5-14B-Instruct	City Locations	0.92	0.01	0.07	0.00	0.01	0.89	0.09	0.00	0.52	0.11	0.37	0.00
	Medical Indications	0.78	0.12	0.10	0.00	0.11	0.80	0.09	0.00	0.40	0.49	0.10	0.00
	Word Definitions	0.81	0.09	0.10	0.00	0.08	0.85	0.07	0.00	0.31	0.62	0.07	0.00
Qwen-2.5-7B	City Locations	0.94	0.02	0.05	0.00	0.03	0.90	0.07	0.00	0.40	0.52	0.08	0.00
	Medical Indications	0.72	0.12	0.17	0.00	0.10	0.78	0.13	0.00	0.34	0.43	0.23	0.00
	Word Definitions	0.72	0.08	0.19	0.00	0.08	0.74	0.19	0.00	0.34	0.41	0.25	0.00
Qwen-2.5-7B-Instruct	City Locations	0.92	0.02	0.07	0.00	0.03	0.91	0.07	0.00	0.50	0.46	0.04	0.00
	Medical Indications	0.76	0.09	0.15	0.00	0.11	0.78	0.11	0.00	0.64	0.23	0.13	0.00
	Word Definitions	0.84	0.10	0.07	0.00	0.11	0.82	0.07	0.00	0.47	0.42	0.11	0.00

Table 18: **Row-wise confusion matrices for the multiclass `sAwMIL` across all ⟨model-dataset pairs⟩.** Each row corresponds to a specific model and a dataset. Columns are grouped by the ground-truth labels (*True*, *False*, *Neither*) with groups of subcolumns that specify the distribution of predictions (*true*, *false*, *neither*, *abstain*). For each statement in a dataset, the predicted class is the class with the highest probability (as estimated by multiclass `sAwMIL`). The values in each group of four subcolumns sum to 1 because they are normalized counts. For example, in the first row under the *True* ground-truth label, we see that *true* predictions have the value of 0.80 – that means that 80% of all the true statements are classified as true. In other words, each row is a flattened (and normalized) confusion matrix.

	True Labels →	True				False				Neither			
	Predicted →	True	False	Neither	Abstain	True	False	Neither	Abstain	True	False	Neither	Abstain
Model ↓	Data Set ↓												
Bio-Medical-Llama-3-8B	City Locations	0.80	0.01	0.00	0.18	0.00	0.88	0.00	0.11	0.00	0.00	1.00	0.00
	Medical Indications	0.77	0.10	0.00	0.12	0.10	0.81	0.01	0.08	0.00	0.00	1.00	0.00
	Word Definitions	0.86	0.09	0.01	0.03	0.11	0.87	0.01	0.02	0.00	0.01	0.98	0.01
Gemma-2-9B	City Locations	0.87	0.00	0.00	0.13	0.00	0.85	0.00	0.15	0.00	0.00	1.00	0.00
	Medical Indications	0.81	0.08	0.01	0.11	0.10	0.73	0.01	0.16	0.01	0.00	0.97	0.02
	Word Definitions	0.86	0.10	0.00	0.04	0.11	0.85	0.01	0.03	0.01	0.01	0.98	0.00
Gemma-2-9B-it	City Locations	0.85	0.02	0.02	0.12	0.02	0.87	0.00	0.11	0.00	0.00	0.98	0.02
	Medical Indications	0.76	0.10	0.01	0.13	0.07	0.84	0.01	0.09	0.01	0.00	0.99	0.01
	Word Definitions	0.83	0.08	0.00	0.08	0.07	0.88	0.01	0.05	0.00	0.01	0.98	0.01
Gemma-7B	City Locations	0.86	0.01	0.00	0.13	0.01	0.89	0.00	0.11	0.00	0.00	1.00	0.00
	Medical Indications	0.75	0.09	0.00	0.17	0.08	0.73	0.01	0.18	0.01	0.01	0.95	0.04
	Word Definitions	0.82	0.11	0.04	0.02	0.15	0.81	0.01	0.03	0.02	0.01	0.95	0.01
Gemma-7B-it	City Locations	0.86	0.02	0.01	0.11	0.02	0.87	0.00	0.11	0.00	0.00	0.99	0.01
	Medical Indications	0.58	0.05	0.01	0.35	0.15	0.50	0.00	0.35	0.00	0.01	0.94	0.05
	Word Definitions	0.69	0.16	0.04	0.11	0.12	0.77	0.03	0.07	0.01	0.01	0.96	0.01
Llama-3-8B	City Locations	0.87	0.01	0.00	0.12	0.01	0.85	0.00	0.14	0.00	0.00	1.00	0.00
	Medical Indications	0.78	0.09	0.01	0.12	0.11	0.77	0.01	0.11	0.00	0.01	0.98	0.01
	Word Definitions	0.87	0.12	0.00	0.00	0.13	0.85	0.01	0.00	0.01	0.01	0.97	0.00
Llama-3.1-8B-Instruct	City Locations	0.87	0.02	0.00	0.11	0.02	0.86	0.00	0.13	0.00	0.00	1.00	0.00
	Medical Indications	0.80	0.12	0.00	0.09	0.09	0.85	0.00	0.06	0.00	0.00	1.00	0.00
	Word Definitions	0.87	0.06	0.00	0.07	0.07	0.86	0.00	0.06	0.00	0.01	0.98	0.01
Llama-3.2-3B	City Locations	0.84	0.03	0.00	0.12	0.04	0.86	0.01	0.09	0.00	0.01	0.99	0.01
	Medical Indications	0.71	0.10	0.00	0.19	0.09	0.70	0.01	0.20	0.00	0.00	0.98	0.02
	Word Definitions	0.74	0.14	0.02	0.10	0.11	0.77	0.02	0.10	0.01	0.01	0.96	0.02
Llama-3.2-3B-Instruct	City Locations	0.86	0.02	0.00	0.11	0.02	0.82	0.00	0.16	0.00	0.00	1.00	0.00
	Medical Indications	0.67	0.12	0.00	0.21	0.11	0.72	0.00	0.18	0.00	0.00	0.99	0.01
	Word Definitions	0.86	0.11	0.03	0.00	0.14	0.85	0.01	0.00	0.01	0.01	0.98	0.00
Llama3-Med42-8B	City Locations	0.87	0.01	0.00	0.12	0.01	0.84	0.00	0.15	0.00	0.00	0.99	0.01
	Medical Indications	0.81	0.10	0.00	0.10	0.11	0.82	0.00	0.07	0.00	0.00	1.00	0.00
	Word Definitions	0.86	0.06	0.01	0.06	0.08	0.85	0.00	0.07	0.01	0.01	0.97	0.02
Mistral-7B-Instruct-v0.3	City Locations	0.86	0.01	0.00	0.13	0.01	0.79	0.00	0.19	0.00	0.00	1.00	0.00
	Medical Indications	0.77	0.09	0.00	0.14	0.10	0.76	0.00	0.14	0.00	0.01	0.99	0.01
	Word Definitions	0.84	0.08	0.03	0.05	0.09	0.87	0.01	0.03	0.01	0.01	0.98	0.01
Mistral-7B-v0.3	City Locations	0.86	0.01	0.00	0.13	0.01	0.86	0.00	0.13	0.00	0.00	1.00	0.00
	Medical Indications	0.75	0.08	0.00	0.17	0.09	0.76	0.00	0.15	0.00	0.00	0.99	0.01
	Word Definitions	0.87	0.11	0.02	0.00	0.14	0.85	0.01	0.00	0.01	0.01	0.98	0.00
Qwen-2.5-14B	City Locations	0.87	0.01	0.00	0.12	0.01	0.86	0.00	0.13	0.00	0.00	0.99	0.01
	Medical Indications	0.78	0.12	0.00	0.10	0.11	0.76	0.00	0.13	0.00	0.00	0.99	0.00
	Word Definitions	0.86	0.12	0.02	0.00	0.13	0.86	0.01	0.00	0.01	0.02	0.98	0.00
Qwen-2.5-14B-Instruct	City Locations	0.82	0.01	0.00	0.17	0.01	0.87	0.00	0.11	0.00	0.00	1.00	0.00
	Medical Indications	0.80	0.12	0.01	0.07	0.07	0.86	0.01	0.07	0.00	0.00	0.99	0.00
	Word Definitions	0.86	0.07	0.01	0.05	0.06	0.86	0.00	0.08	0.00	0.01	0.98	0.01
Qwen-2.5-7B	City Locations	0.84	0.01	0.00	0.16	0.01	0.84	0.00	0.14	0.00	0.00	1.00	0.00
	Medical Indications	0.74	0.10	0.01	0.15	0.10	0.77	0.01	0.13	0.00	0.00	0.98	0.01
	Word Definitions	0.83	0.12	0.02	0.03	0.10	0.87	0.01	0.03	0.01	0.01	0.97	0.01
Qwen-2.5-7B-Instruct	City Locations	0.82	0.02	0.01	0.15	0.02	0.85	0.00	0.13	0.00	0.00	0.99	0.01
	Medical Indications	0.78	0.08	0.01	0.13	0.14	0.72	0.02	0.12	0.01	0.01	0.98	0.01
	Word Definitions	0.83	0.13	0.02	0.02	0.09	0.87	0.02	0.02	0.01	0.01	0.97	0.01

L Code Availability

The code is available at anonymous.4open.science/r/tot-4122/.