

# Label Forensics: Interpreting Hard Labels in Black-Box Text Classifier

Mengyao Du

dumengyao@nudt.edu.cn  
National University of Defense  
Technology  
Changsha, China

Gang Yang

ygang@u.nus.edu  
National University of Singapore  
Singapore

Han Fang

fanghan@nus.edu.sg  
National University of Singapore  
Singapore

Quanjun Yin

yin\_quanjun@163.com  
National University of Defense  
Technology  
Changsha, China

Ee-Chien Chang

changech@comp.nus.edu.sg  
National University of Singapore  
Singapore

## Abstract

The widespread adoption of natural language processing techniques has led to an unprecedented growth of text classifiers across the modern web. Yet many of these models circulate with their internal semantics undocumented or even intentionally withheld. Such opaque classifiers, which may expose only hard-label outputs, can operate in unregulated web environments or be repurposed for unknown intents, raising legitimate forensic and auditing concerns. In this paper, we position ourselves as investigators and work to infer the semantic concept each label encodes in an undocumented black-box classifier.

Specifically, we introduce label forensics, a black-box framework that reconstructs a label's semantic meaning. Concretely, we represent a label by a sentence embedding distribution from which any sample reliably reflects the concept the classifier has implicitly learned for that label. We believe this distribution should maintain two key properties: *precise*, with samples consistently classified into the target label, and *general*, covering the label's broad semantic space. To realize this, we design a semantic neighborhood sampler and an iterative optimization procedure to select representative seed sentences that jointly maximize label consistency and distributional coverage. The final output, an optimized seed sentence set combined with the sampler, constitutes the empirical distribution representing the label's semantics. Experiments on multiple black-box classifiers achieves an average label consistency of around 92.24%, demonstrating that the embedding regions accurately capture each classifier's label semantics. We further validate our framework on an undocumented HuggingFace classifier, with the resulting analysis also presented in this paper, enabling fine-grained label interpretation and supporting responsible AI auditing.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference acronym 'XX, Woodstock, NY*

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-XXXX-X/2018/06  
<https://doi.org/XXXXXXX.XXXXXXX>

## CCS Concepts

- Computing methodologies → Natural language processing; Machine learning approaches.

## Keywords

Model Forensics, Responsible AI Auditing, Model Interpretability

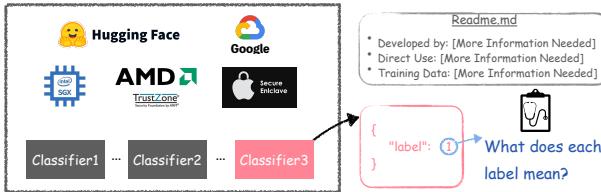
## ACM Reference Format:

Mengyao Du, Gang Yang, Han Fang, Quanjun Yin, and Ee-Chien Chang. 2018. Label Forensics: Interpreting Hard Labels in Black-Box Text Classifier. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 Introduction

Recent advances in NLP (Natural Language Processing) have led to the emergence of a wide range of powerful text classifiers [12, 29, 47]. Many of these models are deployed on modern web platforms for tasks such as sentiment analysis [52], spam filtering [7], and toxicity detection [53]. These models have greatly facilitated real-world applications by enabling automated content moderation [26], personalized recommendations [43], and safety monitoring [40], and are increasingly accessed through lightweight mechanisms such as inference interfaces or cloud-based deployments to achieve inference without maintaining local infrastructure [10, 13, 25, 27].

While many production-grade classifiers are well-documented and designed for legitimate use, a substantial number of models circulating on the web provide no documentation and offer little visibility into their intended purpose. These opaque classifiers often reveal only a hard-label output, as illustrated in Figure 1, without exposing the semantics behind their predictions. Such lack of transparency creates risks in web environments where classifier outputs increasingly influence online interactions, civic engagement, and content governance. Such lack of transparency raises concerns about hidden biases, unintended misuse, or unknown decision logic. From the perspective of safe and responsible AI, this motivates a need for systematic forensic analysis capable of recovering the semantic meaning behind each output label, thereby supporting safe, interpretable, and accountable operation of text classification systems in real-world settings.



**Figure 1: Application scenario of Label Forensics.** Web-based text classifiers often return only JSON hard labels, motivating the need to investigate what these labels actually represent.

In this work, we take the role of an investigator, aiming to understand what a black-box text classifier is truly doing. Given only a text classifier, without documentation, label descriptions, or access to internal signals, we seek to recover the semantic distribution associated with each output label. Concretely, for every label, we aim to characterize the set of natural language sentences that the classifier consistently associates with that label, forming an empirical embedding distribution that reflects its underlying semantic concept.

Achieving this goal is far from straightforward, especially under realistic black-box conditions where the classifier returns only a one-hot label for any input. From a distributional perspective, two key challenges arise. First, the model produces a label for every query, so random probing or isolated triggers offer little insight into what truly defines a class. Second, recovering a label’s semantic distribution requires identifying a set of representative sentences that collectively capture its conceptual meaning, rather than just finding individual inputs that elicit the label. This goal is fundamentally different from prior black-box analysis methods, such as membership inference [20, 23, 33], model inversion [31, 34], and training data extraction [4, 48], which focus on leaking or reconstructing specific training samples. Those techniques operate at the instance level, whereas our task demands distribution-level semantic recovery, which demands methods that can capture broader and more systematic semantic patterns. Filling this gap is crucial for understanding what a classifier’s labels represent and for conducting trustworthy forensic analysis.

To address this challenge, we propose **Label Forensics**, a black-box framework for reconstructing the semantic distribution of each label. We first define this distribution as a semantic region described by the set of natural language sentences that reflect the meaning of the label, along with the broader semantic space associated with them. This semantic region serves as an empirical characterization of the label’s underlying concept. We consider a good distribution to be precise, with samples reliably classified into the target label, and general, covering the broad semantic space of that label. These two properties ensure that the recovered distribution not only mirrors the classifier’s labeling behavior but also provides a faithful and interpretable representation of the underlying concept.

Our approach combines a broad candidate generator, a semantic neighborhood sampler, and an iterative search procedure to reconstruct the distribution of each label. We begin by assembling a diverse pool of label seed sentence sets intended to cover a wide

range of linguistic expressions. The sampler, instantiated as a prefix-tuned encoder-decoder model which we train within our framework, generates local semantic variations around a seed sentence while preserving label consistency. The iterative search then refines a set of representative seeds by maximizing the objective, which jointly measures precision and coverage. Starting from an initial seed, the search expands neighborhoods, filters label-consistent candidates, and incrementally builds an optimized seed set. The resulting seeds, together with the sampler, constitute the empirical distribution for each label, enabling faithful, interpretable, and distribution-level forensics in black-box settings. Experiments on real-world text classifiers, including emotion, sentiment, spam, and jailbreak detection, demonstrate that label forensics yields precise and general semantic summaries, supporting scalable model auditing under opaque deployment. All code and experiments are released at an anonymized repository<sup>1</sup>.

#### Contributions:

- We highlight the importance of black-box text classifier forensics, which aims to recover the semantic meaning behind each output label under hard-label and minimal-access settings, forming a principled foundation for auditing opaque classifiers.
- We propose label forensics, a general-purpose framework that reconstructs label semantics as empirical distributions over natural language sentences. The method combines a prefix-tuned semantic neighborhood sampler with a geometric optimization objective to ensure that the resulting distributions are both *precise* and *general*, enabling faithful and interpretable semantic recovery.
- We evaluate our approach across five text classifiers spanning diverse NLP tasks, as well as a real-world undocumented HuggingFace model. Results show that label forensics consistently reveals label semantics, detects mismatches between declared and learned behaviors, and supports responsible AI auditing.

## 2 Related Works

While several attack techniques in NLP privacy and security inspire components of our pipeline [1, 9, 14], our goal fundamentally differs from theirs. Membership inference attacks [8, 15, 41] and embedding inversion [6, 30] demonstrate that sensitive data can leak through language model outputs. Other efforts develop targeted data-reconstruction attacks aimed at recovering representative training inputs from text classifiers [17]. In addition to these reconstruction-oriented methods, training data extraction attacks, originally developed for generative models, have demonstrated that it is possible to scale up the recovery of training content from deployed models [4, 11, 35]. While our method includes data generation for semantic analysis, our objective is not to reconstruct specific training samples for privacy leakage. Instead, we aim to generate sparse and interpretable samples that support the recovery of precise and general label concepts, rather than reproducing individual training instances.

The interpretability or explainability of language models also aligns with our task [16, 21, 44, 46]. Recent studies have explored

<sup>1</sup>[https://anonymous.4open.science/r/Label\\_Forensic-EED](https://anonymous.4open.science/r/Label_Forensic-EED)

internal mechanisms within language models using sparse autoencoders to extract thousands of human-interpretable features [24, 36], or applying causal interventions to identify robust causal pathways for prediction [22, 28]. Beyond internal probing, black-box interpretability methods have also emerged [5, 37, 38, 42, 45]. Existing approaches usually explain a model behavior by analyzing its inputs and outputs, highlighting important tokens, or approximating the local decision boundary around individual examples. While these methods improve transparency, they mostly clarify why a single prediction was made rather than revealing the broader semantic patterns or label-level structures that the model has learned.

Auditing language models in opaque settings has focused on probing functional behaviors such as refusal patterns [2, 3], jailbreak susceptibility [49], or knowledge tracing [32]. These approaches evaluate how the model behaves on harmful or safety-critical tasks. However, they do not examine what each output label actually represents. In contrast, label forensics focuses on the semantic content of individual labels, aiming to recover the underlying concept a classifier associates with each label rather than assessing its high-level behavioral properties.

While the task of label forensics is underexplored in NLP, conceptually related efforts in computer vision, such as label inference attacks and domain discovery [18, 50, 51]. Yet natural language inputs differ significantly from images: they are discrete, compositional, and semantically entangled, which makes direct transfer of such techniques non-trivial.

### 3 Problem Formulation

#### 3.1 Capabilities of the Investigator

Consider an investigator with black-box and hard-label only access to a text classification model  $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ , where  $\mathcal{X}$  denotes the space of natural language sentences and  $\mathcal{Y}$  is a finite set of class labels. The model  $\mathcal{M}$  is accessible only via queries, as investigators have no access to model internal parameters, architectural details, or gradients. Moreover,  $\mathcal{M}$  returns hard labels only, which do not expose confidence scores, probability distributions, or logits. All queries must be issued in the form of natural language sentences.

The investigator is equipped with several auxiliary resources. An open-source large language model  $\mathcal{G}$  (e.g., Llama or the Qwen series) provides a flexible mechanism for producing diverse natural language sentences. An external corpus offers additional knowledge, including a lexical hierarchy  $\mathcal{W}$  derived from WordNet and a collection of harmful or safety-related sentences, which provides coverage of linguistic regions that are difficult to obtain through  $\mathcal{G}$  alone. Finally, the investigator has a sampler  $\mathcal{S}$  implemented as a prompt-free encoder-decoder pair ( $\text{Enc}, \text{Dec}$ ), which maps a sentence to a continuous semantic embedding and stochastically decodes embeddings back into natural language.

#### 3.2 Goal of Investigation

For each label  $y \in \mathcal{Y}$ , the investigator aims to recover the semantic concept that the classifier  $\mathcal{M}$  implicitly associates with that label. We formalize this concept as a semantic distribution  $\mathbb{D}_y$  over natural language sentences or their embeddings, representing the region of meaning for which the classifier consistently predicts  $y$ .

We formalize the task as approximating a semantic distribution  $\mathbb{D}_y = \{(e_1, \alpha_1), \dots, (e_m, \alpha_m)\}$ , where each embedding  $e_i$  corresponds to a representative sentence associated with label  $y$  and each  $\alpha_i$  characterizes the size of the semantic neighborhood around  $e_i$  within which  $\mathcal{M}$  continues to predict the same label. Intuitively, the distribution  $\mathbb{D}_y$  defines a region of natural language expressions that are both consistently classified as label  $y$  and semantically representative of the label.

A well-formed distribution  $\mathbb{D}_y$  should satisfy two key properties:

- **Precise:** Sentences sampled from  $\mathbb{D}_y$  must be reliably classified by  $\mathcal{M}$  as label  $y$ :

$$\forall x \in \text{supp } \mathbb{D}_y, \quad \mathcal{M}(x) = y. \quad (1)$$

- **General:** The embeddings in  $\mathbb{D}_y$  should form a broad and coherent semantic region. Formally, we define  $\mathbb{E}_{e_i, e_j \in \mathbb{D}_y} [d(e_i, e_j)]$ , where  $d$  for all  $y' \neq y$  where  $d$  denotes a distance measure in the embedding space, and a larger value indicates a wider semantic spread.

#### 3.3 Objective Function

For each label  $y$ , we operationalize these two properties by optimizing the following objective:

$$\max_{\mathbb{D}_y} \Pr_{x \in \text{supp } \mathbb{D}_y} [\mathcal{M}(x) = y] + \lambda \mathbb{E}_{e_i, e_j \in \mathbb{D}_y} [d(e_i, e_j)]. \quad (2)$$

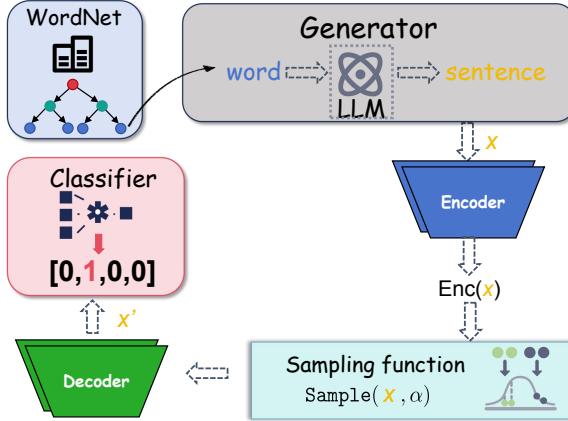
The two terms jointly ensure that  $\mathbb{D}_y$  remains both decision-aligned and semantically expansive, yielding a distribution that reflects the full concept associated with label  $y$ .

### 4 Proposed Framework

The label forensics pipeline consists of three stages: semantic distribution initialization, prompt-free sampler construction with iterative optimization, and semantic interpretation. Figure 2 provides a simplified illustration of the process for constructing a label-specific semantic distribution. We begin by generating a candidate prototype sentence set  $\mathcal{A}_y$  for each label using an open-source LLM conditioned on lexical anchors extracted from corpora such as WordNet. This forms the initial semantic support for label  $y$ . To approximate the distribution  $\mathbb{D}_y$ , we design a sampling mechanism based on  $\mathcal{A}_y$ . The sampling mechanism uses a prompt-free encoder-decoder sampler ( $\text{Enc}, \text{Dec}$ ) where the encoder  $\text{Enc}$  maps  $x \in \mathcal{A}_y$  to a continuous embedding  $e$ . The decoder  $\text{Dec}$  reconstructs a rephrased variant  $x' = \text{Dec}(e)$  from the latent embedding.

#### 4.1 Semantic Distribution Initialization

Recovering the semantic distribution  $\mathbb{D}_y$  for a label  $y$  involves identifying a set of natural language sentences that are both consistently classified as  $y$  and semantically representative of the class. Unlike prior work on data reconstruction attacks, these sentences are not intended to replicate training examples, but rather to capture the underlying semantics of each label. The initialization process begins by collecting a diverse set of prototype sentences and constructing a latent-space sampling mechanism. In the following, we elaborate on each component in detail, describing how the sampler is optimized and how the resulting samples are used to recover the semantic distribution.



**Figure 2: The semantic distribution construction pipeline.** Sentences are generated from WordNet using an LLM. An encoder-decoder architecture serves as the semantic neighborhood sampler. Each seed sentence is paired with a sampling radius  $\alpha$  to define a controllable semantic distribution.

**4.1.1 Heuristic Hierarchical Traversal over WordNet.** To obtain an initial pool of semantically prototype sentences, lexical words are first extracted from WordNet, a structured semantic graph composed of synsets and hypernymy relations. Beginning from root-level abstract words, a breadth-first traversal explores increasingly specific word-level nodes. For each anchor word  $w$ , a sentence  $x \sim \mathcal{G}(\pi(w))$  is generated using a language model  $\mathcal{G}$  under a fixed prompt template  $\pi(\cdot)$ . The sentence is submitted to the black-box classifier  $\mathcal{M}$ , and retained as part of the candidate prototype sentence set  $\mathcal{A}_y$  if  $\mathcal{M}(x) = y$ . This procedure yields a class-conditional set of diverse anchor sentences:

$$\mathcal{A}_y = \{x \sim \mathcal{G}(\pi(w)) \mid w \in \mathcal{W}, \mathcal{M}(x) = y\}, \quad (3)$$

While the WordNet-based initialization is effective for most classes, certain labels such as *jailbreak* remain underrepresented due to class imbalance. To mitigate this issue, a three-path expansion strategy is adopted:

- **Word-level Expansion.** For each word  $w$  associated with the candidate prototype  $x \in \mathcal{A}_y$ ,  $w'$  are retrieved using WordNet relations such as *similar\_to* and *also\_see*. Each  $w'$  is passed to the generator  $\mathcal{G}$  to produce a new sentence  $x' = \mathcal{G}(\pi(w'))$ . If  $\mathcal{M}(x') = y$ , the sample  $x'$  is added to  $\mathcal{A}_y$ .
- **Sentence-level Expansion.** Each  $x \in \mathcal{A}_y$  is paraphrased into a set  $x'_i$  using  $\mathcal{G}$  to explore semantically equivalent sentences. Those satisfying  $\mathcal{M}(x'_i) = y$  are retained.
- **External Corpus Expansion.** A curated corpus of harmful-related sentences is additionally included to cover toxic regions as the LLM  $\mathcal{G}$  is hard to generate toxic or harmful content. Sentences classified as  $y$  are added to  $\mathcal{A}_y$ .

**4.1.2 Prompt-free Sampler Construction.** To explore the semantic distribution  $\mathbb{D}_y$ , we construct a latent-space neighborhood sampler based on a prefix-tuned encoder-decoder architecture.<sup>2</sup> The

<sup>2</sup>Decoder-only models such as GPT lack encoder cross-attention and cannot condition directly on external embeddings.

sampling mechanism uses a prompt-free sampler ( $\text{Enc}, \text{Dec}$ ) where the encoder  $\text{Enc}$  maps  $x \in \mathcal{A}_y$  to a continuous embedding  $e = \text{Enc}(x) \in \mathbb{R}^{l \times d}$ , where  $l$  is the sequence length of  $x$  and  $d$  is the embedding dimension. The decoder  $\text{Dec}$  reconstructs a rephrased variant  $x' = \text{Dec}(e)$  from the latent embedding. To draw samples  $x' \sim \mathbb{D}_y$ , we first sample a candidate prototype sentence uniformly from  $\mathcal{A}_y$  then apply Gaussian noise in its embedding space so that the output sentence inherits the semantic concept but differs in concrete expression. The stochastic sampling function is defined as:

$$x' = \text{Dec}(\text{Enc}(x) + \delta), \quad (4)$$

$$x \sim \mathcal{U}(\mathcal{A}_y), \quad \delta \sim \mathcal{N}(0, \alpha^2 I).$$

Here,  $\mathcal{U}(\cdot)$  means uniformly sampling and  $\alpha$  controls the scale of perturbation in embedding space.

A key requirement is prompt-free generation: the decoder must generate solely from embeddings of prototype sentences, without reliance on handcrafted prompts. However, the standard T5 models rely on textual prompts (e.g., *paraphrase: I am happy*) to control task-oriented generation. To remove this dependency, we fine-tune T5 using prefix tuning, optimizing only a small set of continuous virtual prefix vectors while keeping the backbone frozen. Specifically, we build on the publicly available model<sup>3</sup> and fine-tune it on the ChatGPT-generated dataset<sup>4</sup> to mimic prompt-based behavior in a prompt-free setting.

The training objective aligns the prompt-free and prompt-based models using the following distillation loss:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{x \sim \mathcal{D}} [\text{CE}(\mathcal{T}(\cdot | \pi(x)), \mathcal{T}_{\text{prefix}}(\cdot | x))], \quad (5)$$

where  $\mathcal{T}$  is the original T5 model with prompt  $\pi(x)$ , and  $\mathcal{T}_{\text{prefix}}$  is the prefix-tuned variant without prompts. The learned prefix enables fluent, semantically consistent decoding from latent inputs, and serves as the backbone for controlled perturbation-based sampling.

**4.1.3 Sampling Radius Estimation.** The *sampling radius*  $\alpha$  characterizes the maximum semantic perturbation under which the model consistently classifies a prototype sentence  $s$  with probability at least  $\eta$ . Formally, it is defined as:

$$\alpha(s, \eta) = \sup \left\{ \alpha \mid \mathbb{P}_{\delta \sim \mathcal{N}(0, \alpha^2 I)} [\mathcal{M}(\text{Dec}(\text{Enc}(s) + \delta))] = \mathcal{M}(s) \right\} \geq \eta. \quad (6)$$

Here,  $\eta \in (0, 1)$  denotes a confidence threshold (e.g.,  $\eta = 0.7$ ), ensuring that sampled variants retain the original label with high probability. A large  $\alpha(s, \eta)$  indicates that the prototype lies at the center of a semantically stable region: the classifier's output remains consistent even under strong latent perturbations.

The sampling radius  $\alpha(s, \eta)$  is estimated via Monte Carlo sampling with binary search: for each candidate  $\alpha$ , we draw  $m$  perturbations from  $\mathcal{N}(0, \alpha^2 I)$ , decode each perturbed embedding, and query the classifier. If the label match rate exceeds  $\eta$ , the search radius is expanded; otherwise, it is reduced. This continues until convergence, yielding a robustness estimate  $\alpha(s, \eta)$  per sentence (see Listing 1).

<sup>3</sup>[https://huggingface.co/humarin/chatgpt\\_paraphraser\\_on\\_T5\\_base](https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base)

<sup>4</sup><https://huggingface.co/datasets/humarin/chatgpt-paraphrases>

Listing 1: Estimating sampling radius via binary search

---

```

1 def estimate_alpha(s, eta, m):
2     Z = Enc(s); y = M(Dec(Z))
3     low, high = alpha_min, alpha_max
4     eps = 1e-3
5     while high - low > eps:
6         alpha = (low + high) / 2
7         count = 0
8         for _ in range(m):
9             delta = Gaussian(std=alpha)
10            y_hat = M(Dec(Z + delta))
11            count += (y_hat == y)
12            if count / m >= eta:
13                low = alpha
14            else:
15                high = alpha
16    return low

```

---

## 4.2 Prototype Scoring Objective

To recover a precise and general distribution  $\mathbb{D}_y$  for each class  $y$ , we introduce a scoring objective to optimize prototype sentences  $s \in \mathcal{A}_y$ . In addition to the sampling radius  $\alpha(s, \eta)$ , which reflects classification consistency under perturbations, we incorporate two embedding-based criteria that further identify semantically representative samples: *consistency* and *separability*.

Specifically, let  $\mathbf{z} \in \mathbb{R}^d$  denote the sentence embedding for  $s$ , computed via mean pooling over the final encoder hidden states  $\mathbf{z} = \frac{1}{l} \sum_{i=1}^l \mathbf{h}_i$ , where  $\{\mathbf{h}_i\}_{i=1}^L$  are the token embeddings from the encoder  $\text{Enc}$ . Given a predicted label  $y$ , the centroid of class  $\mathbf{c}_y$  is computed as the mean of normalized sentence embeddings in  $\mathcal{A}_y$ . We define:

- **Consistency:** Measures how well a sample aligns with the core semantics of its predicted class, defined as the cosine similarity between  $\mathbf{z}$  and its class centroid  $\mathbf{c}_y$ ;
- **Separability:** Penalizes proximity defined as the inverse of the maximum cosine similarity between  $\mathbf{z}$  and any other class centroid  $\mathbf{c}_{\tilde{y}}$ , where  $\tilde{y} \in \mathcal{Y} \setminus \{y\}$ .

These two metrics are designed to ensure that selected samples are semantically faithful to their class while remaining distinct from others. The overall scoring function is:

$$\mathcal{R}(s) = \alpha(s, \eta) + \lambda \cdot \cos(\mathbf{z}, \mathbf{c}_y) + \gamma \cdot \left(1 - \max_{\tilde{y} \in \mathcal{Y} \setminus \{y\}} \cos(\mathbf{z}, \mathbf{c}_{\tilde{y}})\right), \quad (7)$$

where  $\lambda, \gamma \geq 0$  control the balance between consistency and separability.

The optimization objective is to select a subset of  $K$  prototype sentences  $\mathcal{S}_y \subset \mathcal{A}_y$  that maximizes the total representativeness:

$$\mathcal{S}_y^* = \arg \max_{|\mathcal{S}_y|=K} \sum_{s \in \mathcal{S}_y} \mathcal{R}(s). \quad (8)$$

The selected subset  $\mathcal{S}_y^*$  provides a compact and interpretable summary of the semantic space associated with label  $y$ , and serves as the support for the induced distribution  $\mathbb{D}_y$ , where each prototype  $s \in \mathcal{S}_y^*$  is associated with a sampling radius  $\alpha(s, \eta)$ .

## 4.3 Interpreting the Semantic Distribution

After constructing the semantic distribution for each label  $y \in \mathcal{Y}$ , the next objective is to distill high-level, interpretable descriptions

that summarize the label's semantics in a human-understandable form. Specifically, the goal is to extract abstract words or phrases that generalize across diverse prototype sentences. For example, in an emotion classifier, rather than listing expressions like *I can't stop smiling* or *What a wonderful day*, we aim to induce canonical concepts such as *joy* that capture the underlying labeling behavior of the model.

As  $\mathbb{D}_y$  encodes the text classifier's behavior across semantically coherent inputs, we can directly prompt a generative language model  $G$  to attribute an interpretable label description. Specifically, the model is asked to generate a set of candidate label descriptions  $\{d_1, \dots, d_k\}$  that summarize the shared semantics of  $\mathcal{S}_y^*$ . Each  $d_i$  is expected to be a short phrase that captures the essence of class  $y$ .

To evaluate the alignment between each candidate  $d_i$  and the semantic behavior of the classifier, we employ a pretrained natural language inference (NLI) model as a scoring function. For each prototype sentence  $s \in \mathcal{S}_y^*$ , we compute the entailment score  $f_{\text{NLI}}(s, d_i)$ . The overall quality of label description is quantified by its entailment hit rate:

$$\mathcal{H}(d_i) = \frac{1}{|\mathcal{S}_y^*|} \sum_{s \in \mathcal{S}_y^*} \mathbb{I}[f_{\text{NLI}}(s, d_i) \geq \tau], \quad (9)$$

where  $\tau$  is a confidence threshold (e.g.,  $\tau = 0.6$ ), and  $\mathbb{I}[\cdot]$  is the indicator function. This score measures how often the prototype sentences entail the candidate label description according to the NLI model. Label descriptions with higher  $\mathcal{H}(d_i)$  are considered more faithful and class-representative. This mechanism provides an interpretable summarization of each label's semantic footprint in terms of human-readable descriptions, bridging the gap between black-box classifier predictions and semantic transparency.

## 5 Experimental Results

### 5.1 Implementation Details

We evaluate our method on five publicly available black-box text classifiers, each addressing a distinct NLP task:

- **Emotion Classification ( $M_{\text{emo}}$ ):** a DistilRoBERTa-based model trained to classify text into one of seven emotion categories.
- **Sentiment Analysis ( $M_{\text{sent}}$ ):** a multilingual BERT model that assigns a sentiment rating from 1 (most negative) to 5 (most positive). Since the model lacks explicit label descriptions, it presents a natural scenario for unsupervised label forensics.
- **Spam Detection ( $M_{\text{spam}}$ ):** a binary classifier trained on corpora of emails and SMS messages to distinguish spam from legitimate content.
- **Jailbreak Detection ( $M_{\text{jail}}$ ):** a DistilBERT model designed to identify adversarial jailbreak prompts.
- **Toxicity Detection ( $M_{\text{tox}}$ ):** a RoBERTa-based classifier for binary toxicity classification.

All models in our study operate under a strict black-box setting, exposing only hard-label predictions without access to confidence scores or logits. To construct prototype sentence candidates, we traverse the lexical graph of WordNet 3.1 using hierarchical heuristics. For probing and concept induction, we utilize the open-source

large language models Qwen-Chat and Llama-3.1-8B-Instruct as the generator  $\mathcal{G}$ , configured with nucleus sampling ( $p = 0.9$ ) and temperature 1.0 to balance output diversity and semantic fidelity.

For training the prompt-free neighborhood sampler, we adopt prefix tuning by appending 20 learnable virtual tokens to both the encoder and decoder. The tuning follows a teacher–student distillation setup and is performed for 2 epochs with a batch size of 64 and a learning rate of  $10^{-4}$ . For each label  $y$ , to assess the semantic alignment of induced concepts, we use an entailment model  $f_{NLI}$ , with the entailment confidence threshold  $\tau$  set to 0.6.

All experiments are conducted on a workstation equipped with 4 NVIDIA A40 GPUs (46 GB VRAM each).

## 5.2 Evaluation Setup

To assess the structural quality of the induced semantic distribution, we evaluate the geometry of prototype sentence embeddings using two metrics: intra-class distance and inter-class distance. All embeddings are derived from the selected prototype subsets  $\mathcal{S}_y^*$  for each class  $y \in \mathcal{Y}$ . Specifically, we compute sentence embeddings  $\mathbf{z} \in \mathbb{R}^D$  using pretrained encoders such as Sentence-BERT and SimCSE to ensure model-agnosticity. Let  $\mathcal{Z}_y = \{\mathbf{z}_1, \dots, \mathbf{z}_{n_y}\}$  denote the embedding set obtained from  $\mathcal{S}_y^*$ . The centroid of each class is defined as:

$$\mathbf{c}_y = \frac{1}{|\mathcal{Z}_y|} \sum_{\mathbf{z} \in \mathcal{Z}_y} \mathbf{z}. \quad (10)$$

The intra-class distance measures the average distance between samples and their class centroid:

$$d_{\text{intra}} = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \left( \frac{1}{|\mathcal{Z}_y|} \sum_{\mathbf{z} \in \mathcal{Z}_y} (1 - \cos(\mathbf{z}, \mathbf{c}_y)) \right). \quad (11)$$

The inter-class distance computes the average distance between all pairs of class centroids:

$$d_{\text{inter}} = \frac{1}{\binom{|\mathcal{Y}|}{2}} \sum_{y_i < y_j} (1 - \cos(\mathbf{c}_{y_i}, \mathbf{c}_{y_j})). \quad (12)$$

Low  $d_{\text{intra}}$  indicates that prototype samples are semantically coherent within each class, while high  $d_{\text{inter}}$  suggests that the induced classes are geometrically well-separated in embedding space.

To evaluate the quality of the generated semantic distributions, we adopt BERTScore and Self-BLEU. BERTScore assesses semantic similarity between generated samples and their corresponding prototypes, capturing meaning beyond surface-level tokens. In contrast, Self-BLEU measures intra-set diversity; lower values indicate greater variation among samples, helping to identify redundancy or mode collapse.

## 5.3 Quantitative Results

**5.3.1 Label Consistency in Reconstructed Embedding Space.** We first assess the fidelity of the reconstructed embedding space. For each label-specific embedding space, we uniformly sample 100 embeddings, decode them into sentences, and evaluate whether the classifier assigns the original label. A higher accuracy indicates that the learned embedding space faithfully captures the classifier’s decision manifold, preserving both local smoothness and label-level semantic coherence. As shown in Table 1, across all five classifiers,

Label	Emotion	Sentiment	Spam	Jailbreak	Toxicity
0	86.0	93.0	99.0	100.0	100.0
1	93.0	86.0	93.0	84.0	91.0
2	95.0	82.0	—	—	—
3	93.0	89.0	—	—	—
4	88.0	83.0	—	—	—
5	91.0	—	—	—	—
6	92.0	—	—	—	—
Avg.	91.1	86.6	96.0	92.0	95.5

**Table 1: Accuracy (%) of classifier predictions on 100 sampled sentences per label generated from the reconstructed embedding space.**

Task	Label	Inferred Concepts
Emotion	Anger	anger, hatred
	Disgust	disgust, discomfort
	Fear	fear, danger
	Joy	joy, cheerfulness
	Neutral	standards, balance
	Sadness	sadness, heartache
Sentiment	Surprise	astonishment, surprise
	Very negative	theft, fraud
	Negative	fatigue, weakened
	Neutral	conventional, tradition
	Positive	effectiveness, effort
Spam	Very positive	successful, happiness
	Ham	brightness, cleanliness
Jailbreak	Spam	irrationality, trustlessness
	Clean Prompt	brightness, relaxation
Toxicity	Jailbreak	fulfillment, prerequisites
	Non-Toxic	responsibility, health
	Toxic	irrationality, harm

**Table 2: Representative label descriptions inferred for each label across five black-box classifiers.**

the recovered embedding regions exhibit strong label consistency. Binary models such as toxicity (95.5%) and spam (96.0%) show almost perfectly coherent embedding neighborhoods, confirming that their decision boundaries are sharp and well-separated. The jailbreak detector yields a mean consistency of 92.0%, though its harmful-content region (Label 1: 84%) is less stable, reflecting the inherently irregular boundary around safety-critical content. Multi-class models show similarly robust behavior. The emotion classifier achieves 91.1% average consistency across seven labels, and the sentiment classifier maintains 86.6% despite its finer-grained five-way label space. These results demonstrate that the embedding reconstruction process successfully recovers classifier-aligned semantic regions that remain internally coherent under sampling-based perturbations.

Task	Model	$d_{\text{intra}} \downarrow$	$d_{\text{inter}} \uparrow$	$r \uparrow$
Emotion	SBERT	0.7766	0.9193	1.18
	SimCSE	0.4766	0.5959	1.25
Sentiment	SBERT	0.8802	0.9376	1.07
	SimCSE	0.5330	0.5954	1.12
Spam	SBERT	0.5210	0.9977	1.92
	SimCSE	0.3455	0.6152	1.78
Jailbreak	SBERT	0.6128	1.0250	1.67
	SimCSE	0.4110	0.6503	1.58
Toxicity	SBERT	0.7632	0.9470	1.24
	SimCSE	0.4941	0.6258	1.27

**Table 3: Intra-class and inter-class distances of sentence embeddings across models. Lower  $d_{\text{intra}}$  and higher  $d_{\text{inter}}$  indicate better geometric structure. The ratio  $r = d_{\text{inter}}/d_{\text{intra}}$  quantifies overall class separability.**

Task	BERTScore $\uparrow$	Consistency $\uparrow$	BLEU $\downarrow$
Emotion	0.94	100.0%	0.22
Sentiment	0.95	100.0%	0.18
Spam	0.93	100.0%	0.23
Jailbreak	0.93	100.0%	0.28
Toxicity	0.94	100.0%	0.16

**Table 4: Evaluation of neighborhood sampler over prototype sentences  $\mathcal{S}_y^*$  across tasks. BERTScore captures semantic fidelity, Consistency measures label preservation, and BLEU reflects lexical diversity.**

**5.3.2 Semantic Alignment with Ground-Truth Labels.** To bridge the gap between semantic distribution and human interpretability, we distill the recovered distributions into concise, natural language label descriptions. Table 2 reports the inferred descriptions across five black-box classifiers spanning diverse NLP tasks.

The quality of inferred label descriptions varies across tasks. For emotion classification, the generated descriptions exhibit nearly one-to-one correspondence with the true emotion categories (e.g., joy, anger, fear), indicating that the semantic distributions accurately capture label semantics. In sentiment analysis, despite the model providing only numeric labels (1–5) without textual descriptions, the induced descriptions align closely with sentiment polarity. For tasks with less well-defined or inherently subjective label semantics, such as spam detection, jailbreak detection, and toxicity classification, the recovered descriptions still exhibit coherent and interpretable behavioral patterns. For example, the spam label is associated with terms such as irrationality and trustlessness, while the toxic label yields descriptors like harm, which align with human intuition. Jailbreak detection, being the least well-defined, produces less precise yet still informative descriptions (e.g., fulfillment, prerequisites), which capture the general intent of adversarial prompts.

**5.3.3 Geometric Analysis of Prototype Sentences.** To evaluate the structural coherence of prototype sentences  $\mathcal{S}_y^*$  in embedding space, we measure intra-class compactness and inter-class separability. These properties reflect whether the selected prototypes form coherent, label-specific regions. We compute intra-class distance  $d_{\text{intra}}$  and inter-class distance  $d_{\text{inter}}$ , defined in Equations 11 and 12, respectively, and use their ratio  $r = d_{\text{inter}}/d_{\text{intra}}$  as a class separability metric:  $r > 1$  suggests meaningful label structure.

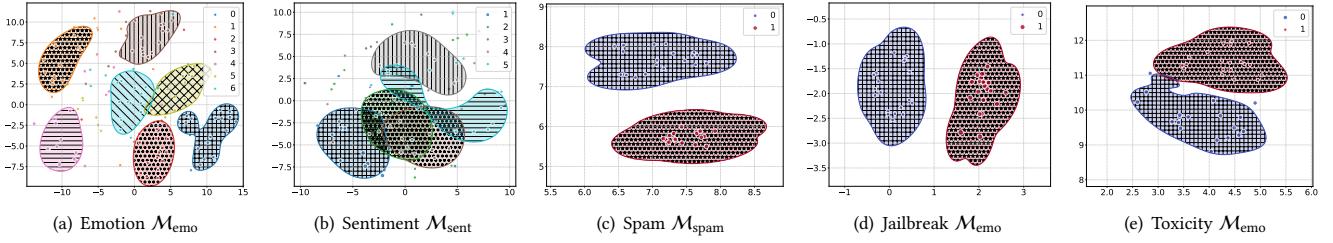
We conduct this analysis using SBERT [39] and SimCSE [19], two widely adopted sentence embedding models trained with contrastive learning, known for their effectiveness in capturing fine-grained semantic similarity. As shown in Table 3, for all text classifiers, the separability ratio  $r$  remains consistently above 1.0. This indicates that the prototypes occupy well-defined regions in embedding space, with intra-class cohesion and inter-class distinction preserved. The highest values of  $r$  are observed in the spam detection and jailbreak detection tasks—reaching up to 1.92 for SBERT, which suggests particularly strong inter-class separability in their latent representations. In contrast, sentiment classification yields the lowest separability ( $r = 1.07$  for SBERT and 1.12 for SimCSE), likely due to the ordinal and overlapping nature of sentiment labels ranging from 1 to 5.

**5.3.4 Visualization of Embedding Geometry.** To qualitatively assess the structure of the recovered semantic distributions, we visualize the prototype sentence embeddings using t-SNE. As shown in Figure 3, each point represents a sentence embedding, with color denoting its predicted label and point size proportional to its estimated sampling radius  $\alpha$ .

Across all classification tasks, the visualizations reveal well-formed clustering patterns: samples within the same label tend to form compact groups, while inter-class regions are largely separable. Notably, the sentiment analysis task exhibits partial overlap between adjacent clusters, consistent with our earlier geometric analysis. This phenomenon reflects the ordinal structure of sentiment labels, which span a graded polarity continuum (e.g., from 1 to 5) rather than mutually exclusive categories. Such overlaps are expected, as semantically similar instances may reside near class boundaries in the embedding space.

**5.3.5 Effectiveness of Prompt-Free Sampler.** We evaluate the quality of the prompt-free sampler by analyzing its outputs over the set of prototype sentences  $\mathcal{S}_y^*$ . Specifically, we assess whether the generated samples preserve the semantics of the original sentence, maintain classification consistency under the target label, and exhibit sufficient diversity to support generalizable concept induction. To quantify semantic preservation, we compute the BERTScore between each seed sentence and its generated variants, capturing the degree of meaning retention. Classification consistency is measured as the percentage of generated sentences that are still classified as label  $y$  by the black-box model  $\mathcal{M}$ , indicating whether the sampler remains within the decision boundary of the intended class. To assess lexical variability, the Self-BLEU (abbreviated as BLEU) is computed, indicating greater divergence from the original phrasing.

As shown in Table 4, the prompt-free sampler consistently produces semantically faithful paraphrases (BERTScore  $\geq 0.93$ ) that retain the original label with perfect classification consistency across all tasks. Moreover, BLEU scores remain below 0.30, indicating that



**Figure 3: t-SNE visualization of prototype sentence embeddings across five classification tasks. Each point represents a sentence, colored by its class label, while shaded regions show the kernel density estimate for each class.**

Label	Representative Constructed Samples
<b>0 (inferred negative semantics)</b>	<i>The company’s new policy is quite nasty to employees.</i> <i>The poet’s cryptical language made it challenging for readers to interpret his work.</i> <i>The politician’s ingratiatory remarks were met with widespread criticism from the public.</i>
<b>1 (inferred positive semantics)</b>	<i>The company will satisfy its customers with the new product launch.</i> <i>His perpetually optimistic attitude made him a joy to be around.</i> <i>After being cooked, the meal was surprisingly delicious.</i>

**Table 5: Constructed example sentences used to examine the semantic tendencies of each label.**

the generated samples avoid collapse and provide diverse instantiations of each concept. These results confirm the effectiveness of the sampler in constructing label-consistent, semantically rich distributions  $\mathbb{D}_y$  that serve as a foundation for downstream analysis.

#### 5.4 Real Case Study: Forensics of an Undocumented HuggingFace Classifier

To demonstrate the practicality of our label-forensics framework, we conduct a real-world case study on the HuggingFace model hub, one of the largest publicly accessible repositories of machine learning models. The platform hosts more than 106,000 text-classification models covering diverse nlp tasks.

Specifically, we select a newly uploaded binary text classifier<sup>5</sup> which contains no documentation, no dataset description, and no explanation of label meanings. Our goal is to construct an embedding representation for each output label and use it to infer both the task identity and the latent meaning of the labels. We begin our analysis by examining the semantic characteristics associated with each label without relying on any reference models, allowing us to observe how the classifier groups different types of inputs.

Using our label-forensics pipeline, we generate paraphrases for label 0 and label 1 and inspect their thematic patterns. As shown in Table 5, the constructed samples display a clear and consistent contrast: sentences assigned to label 0 tend to carry negative or critical tones, while those assigned to label 1 generally express positive or supportive meanings. This pattern is stable across multiple paraphrases and remains visible even when we increase the diversity of the generated samples. Such consistency indicates that the classifier is not assigning these labels arbitrarily but is grouping inputs according to the broad sentiment expressed in the text. Therefore,

we infer that label 0 and label 1 correspond to different sentiment directions, with one leaning negative and the other leaning positive.

To further verify the correctness of our forensic analysis, we compute similarity scores using cosine similarity and compare the target model’s label-specific paraphrase embeddings against all labels of each reference classifier. Table 6 summarizes the closest semantic matches. Both labels of the target model align most strongly with the sentiment classifier, where target label 0 corresponds to *Very Negative* and target label 1 corresponds to *Very Positive*. This confirms the model-level attribution and provides a fine-grained interpretation of its label semantics.

Target Label	Reference Model	Closest Label	Similarity
0	Emotion	Sadness	0.1472
	Sentiment	Very Negative	<b>0.3585</b>
	Spam	Ham	0.3118
	Toxicity	Non-Toxic	0.2760
	Jailbreak	Clean Prompt	0.1559
1	Emotion	Anger	0.2779
	Sentiment	Very Positive	<b>0.3158</b>
	Spam	Ham	0.2357
	Toxicity	Non-Toxic	0.2332
	Jailbreak	Jailbreak	0.2513

**Table 6: Label-level alignment between the target classifier and reference models.**

## 6 Conclusion

The rapid growth of web-scale text classifiers has made understanding their underlying label semantics increasingly important. In this

<sup>5</sup><https://huggingface.co/sanderhs1/trained-mydata>

paper, we introduce a label forensics framework that reconstructs the semantic behavior of a text classifier. Our framework consists of three components: a WordNet-based initialization that builds an anchor pool for each label, a prompt-free encoder–decoder sampler that expands these anchors into a label-specific sentence distribution, and a semantic interpretation module that derives coherent label meanings from the reconstructed distributions. Using this pipeline, our method enables investigators to reason about a classifier’s true behavior using only hard-label outputs. Experiments across five representative classifiers demonstrate that the recovered embedding regions achieve an average label consistency of 92.24%, faithfully capturing decision-level semantics and exhibiting strong stability under sampling-based perturbations.

Beyond benchmark models, we further validate the framework on an undocumented HuggingFace classifier, showing that label forensics can reliably infer task identity and latent label meanings even in the absence of any documentation. This underscores the practical value of our approach for auditing real-world deployed models and supporting more transparent and trustworthy Web AI systems.

## References

- [1] Francisco Aguilera-Martínez and Fernando Berzal. 2025. LLM Security: Vulnerabilities, Attacks, Defenses, and Countermeasures. *arXiv preprint arXiv:2505.01177* 1, 1 (2025), 1–10.
- [2] Maryam Amirizaniani, Elias Martin, Tanya Roosta, Aman Chadha, and Chirag Shah. 2024. AuditLLM: A Tool for Auditing Large Language Models Using Multiprobe Approach. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) (CIKM ’24). Association for Computing Machinery, New York, NY, USA, 5174–5179. doi:10.1145/3627673.367922
- [3] Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssy, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems* 37 (2024), 136037–136083.
- [4] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Comyn, Eric Wallace, David Rolnick, and Florian Tramèr. 2024. Stealing part of a production language model. In *ICML*. PMLR. <https://openreview.net/forum?id=VE3yWxt3KB>
- [5] Guangyao Chen, Kai Horstmann, Zhilong Wang, and Fengqi You. 2025. Automated Essential Concept Discovery for Few-Shot Out-of-Distribution Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 3964–3974.
- [6] Yiyi Chen, Heather Lent, and Johannes Bjerva. 2024. Text embedding inversion security for multilingual language models. *arXiv preprint arXiv:2401.12192* (2024).
- [7] Girija Chetty, Hieu Bui, and Matthew White. 2019. Deep learning based spam detection system. In *2019 International Conference on Machine Learning and Data Engineering (icMLDE)*. IEEE, 91–96.
- [8] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International conference on machine learning*. PMLR, 1964–1974.
- [9] Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vinjña Jain, and Aman Chadha. 2024. Breaking down the defenses: A comparative survey of attacks on large language models. *arXiv preprint arXiv:2403.04786* (2024).
- [10] Yunjie Deng, Chenxu Wang, Shunchang Yu, Shiqing Liu, Zhenyu Ning, Kevin Leach, Jin Li, Shoumeng Yan, Zhengyu He, Jiamlong Cao, et al. 2022. Strongbox: A gpu tee on arm endpoints. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 769–783.
- [11] Jérémie Dentan, Arnaud Paran, and Aymen Shabou. 2024. Reconstructing training data from document understanding models. In *33rd USENIX Security Symposium (USENIX Security 24)*. 6813–6830.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.
- [13] Aritra Dhar, Clément Thorens, Lara Magdalena Lazier, and Lukas Cavigelli. 2025. GuardAIn: Protecting Emerging Generative AI Workloads on Heterogeneous NPU. In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 4155–4172.
- [14] Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. Attacks, defenses and evaluations for llm conversation safety: A survey. *arXiv preprint arXiv:2402.09283* (2024).
- [15] Michael Duan, Anshuman Suri, Nilofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841* (2024).
- [16] Upol Ehsan, Elizabeth A Watkins, Philipp Wintersberger, Carina Manger, Sunnie SY Kim, Niels Van Berkelaer, Andreas Riener, and Mark O Riedl. 2024. Human-centered explainable AI (HCXAI): Reloading explainability in the era of large language models (LLMs). In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–6.
- [17] Adel Elmahdy and Ahmed Salem. 2023. Deconstructing classifiers: Towards a data reconstruction attack against text classification models. *arXiv preprint arXiv:2306.13789* (2023).
- [18] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X. Liu, and Ting Wang. 2022. Label Inference Attacks Against Vertical Federated Learning. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1397–1414. <https://www.usenix.org/conference/usenixsecurity22/presentation/fu-chong>
- [19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).
- [20] Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. 2025. Towards label-only membership inference attack against pre-trained large language models. In *USENIX Security*.
- [21] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing Uncertainty for Large Language Models through Input Clarification Ensembling. In *ICML*. <https://openreview.net/forum?id=byxXa99PtF>
- [22] Jing Huang, Junyi Tao, Thomas Icard, Diyi Yang, and Christopher Potts. 2025. Internal Causal Mechanisms Robustly Predict Language Model Out-of-Distribution Behaviors. In *Forty-second International Conference on Machine Learning*. <https://openreview.net/forum?id=Ofa1cspTrv>
- [23] Zhiheng Huang, Yannan Liu, Daojing He, and Yu Li. 2025. DF-MIA: A Distribution-Free Membership Inference Attack on Fine-Tuned Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 343–351.
- [24] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.
- [25] HuggingFace. 2025. HuggingFace Inference API Documentation. <https://huggingface.co/inference-api> Online; accessed 15 February 2025.
- [26] Shagun Jhaver, Quan Ze Chen, Detlef Knauß, and Amy X Zhang. 2022. Designing word filter tools for creator-led comment moderation. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–21.
- [27] Jigsaw and Google. [n. d.] Google Perspective API. <https://perspectiveapi.com>. Accessed: 2025-02-15.
- [28] Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research* (2023).
- [29] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
- [30] Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010* (2023).
- [31] Yu Lin, Qizhi Zhang, Quanwei Cai, Jue Hong, Wu Ye, Huiqi Liu, and Bing Duan. 2024. An inversion attack against obfuscated embedding matrix in language model inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2100–2104.
- [32] Samuel Marks, Johannes Treutlein, Trenton Bricken, Jack Lindsey, Jonathan Marcus, Siddharth Mishra-Sharma, Daniel Ziegler, Emmanuel Ameisen, Joshua Batson, Tim Belonax, et al. 2025. Auditing language models for hidden objectives. *arXiv preprint arXiv:2503.10965* (2025).
- [33] Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. 2025. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it). In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, IEEE, 385–401.
- [34] John Xavier Morris, Wenting Zhao, Justin T Chiu, Vitaly Shmatikov, and Alexander M Rush. 2024. Language Model Inversion. In *The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=19dWHpGkPj>
- [35] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. 2025. Scalable Extraction of Training Data from Aligned, Production Language Models. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=vjel3nWP2a>

- [36] Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2024. Towards interpreting visual information processing in vision-language models. *arXiv preprint arXiv:2410.07149* (2024).
- [37] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. 2023. Label-free Concept Bottleneck Models. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=FlCg47MNvBA>
- [38] Paulius Rauba, Qiyao Wei, and Mihaela van der Schaar. 2025. Auditing language models with distribution-based sensitivity analysis. In *The 28th International Conference on Artificial Intelligence and Statistics*. <https://openreview.net/forum?id=iLNQ2m4GTY>
- [39] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- [40] Sehrish Sharifi, Andrea Stocco, and Lionel C Briand. 2025. System safety monitoring of learned components using temporal metric forecasting. *ACM Transactions on Software Engineering and Methodology* 34, 6 (2025), 1–43.
- [41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.
- [42] Chandan Singh, Aliyah R Hsu, Richard Antonello, Shailee Jain, Alexander G Huth, Bin Yu, and Jianfeng Gao. 2023. Explaining black box text modules in natural language with language models. *arXiv preprint arXiv:2305.09863* (2023).
- [43] Jia Wang, Yungang Feng, Elham Naghizade, Lida Rashidi, Kwan Hui Lim, and Kate Lee. 2018. Happiness is a choice: sentiment and activity-aware location recommendation. In *Companion Proceedings of the The Web Conference 2018*. 1401–1405.
- [44] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems* 36 (2023), 15614–15638.
- [45] Ximing Wen. 2025. Language model meets prototypes: Towards interpretable text classification models through prototypical networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 29307–29308.
- [46] Jingyuan Yang, Dapeng Chen, Yajing Sun, Rongjun Li, Zhiyong Feng, and Wei Peng. 2024. Enhancing Semantic Consistency of Large Language Models through Model Editing: An Interpretability-Oriented Approach. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3343–3353. doi:10.18653/v1/2024.findings-acl.199
- [47] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).
- [48] Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. 2023. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*. PMLR, 40306–40320.
- [49] Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. 2024. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*. 4675–4692.
- [50] Jiyi Zhang, Han Fang, and Ee-Chien Chang. 2024. Finding Input Data Domains of Image Classification Models with Hard-Label Black-Box Access. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 11166–11174.
- [51] Hanyu Zhao, Zijie Pan, Yajie Wang, Zuobin Ying, Lei Xu, and Yu-an Tan. 2025. Personalized Label Inference Attack in Federated Transfer Learning via Contrastive Meta Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 22777–22785.
- [52] Yang Zhao, Tetsuya Nasukawa, Masayasu Muraoka, and Bishwaranjan Bhattacharjee. 2023. A simple yet strong domain-agnostic de-bias method for zero-shot sentiment classification. In *Findings of the Association for Computational Linguistics: ACL 2023*. 3923–3931.
- [53] Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. 2021. A comparative study of using pre-trained language models for toxic comment classification. In *Companion Proceedings of the Web Conference 2021*. 500–507.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009