How to Generalize the Detection of AI-Generated Text: Confounding Neurons

Anonymous ACL submission

Abstract

Detectors of LLM-generated text suffer from poor domain shifts generalization ability. Yet, reliable text detection methods in the wild are of paramount importance for plagiarism detection, integrity of the public discourse, and AI safety. Linguistic and domain confounders introduce spurious correlations, leading to poor out-of-distribution (OOD) performance. In this work we introduce the concept of confounding neurons, individual neurons within transformers-based detectors that encode dataset-specific biases rather than task-specific signals. Leveraging confounding neurons, we propose a novel post-hoc, neuron-level intervention framework to disentangle AI-generated text detection factors from data-specific biases. Through extensive experiments we prove its ability to effectively reduce topic-specific biases, enhancing the model's ability to generalize across domains.

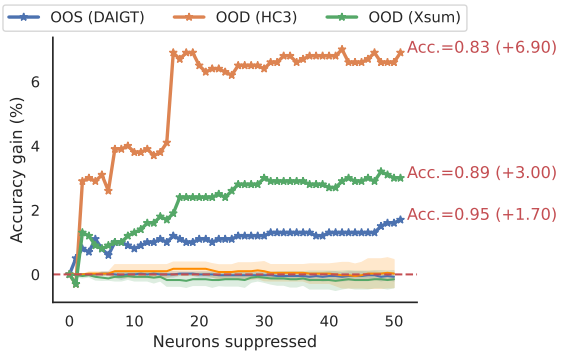

Figure 1: Accuracy gain in a human-vs-LLM detector BERT-based removing the top-K ($K = 1 - 50$) **confounding neurons**. Removing as few as 20 ($\sim 0.05\%$) confounding neurons in the feed-forward MLPs intermediate layers results in up to a 6.9% improvement in the test sets (in-domain, out-of-sample (OOS), and two out-of-distribution (OOD)). The shaded area around zero corresponds to the random baseline.

1 Introduction

The rapid development of Large Language Models (LLMs) has revolutionized natural language processing (NLP), allowing machines to produce text that mirrors human writing in coherence and contextual relevance. However, as LLMs become increasingly sophisticated, identifying AI-generated text poses a critical challenge (Wu et al., 2025a; Zhou and Wang, 2024). This task is particularly pressing in contexts such as academic integrity, misinformation detection, authorship attribution, and cybersecurity, where the misuse of AI-generated content raises ethical and societal issues (Beigi et al., 2024; Gui et al., 2025). Despite advancements in AI detection methods, state-of-the-art (SOTA) systems still exhibit significant generalization failures, especially when applied across diverse domains, languages, and models (Wu et al., 2025a; Gritsai et al., 2024).

Detection methodologies for AI-generated text, based on fine-tuned transformer models (e.g., RoBERTa, XLM-R), achieve high accuracy (>99%) on controlled datasets, their performance on out-of-distribution (OOD) data remains unreliable, limiting their applicability in real-world scenarios (Wu et al., 2025a; Wang et al., 2024b).

Recent benchmarks, such as M4 (Multi-Generator, Multi-Domain, Multi-Lingual) (Wang et al., 2024b) and the GenAI Content Detection Tasks (Lekkala et al., 2025), have underscored the fragility of current detectors, particularly in cross-domain generalization (Xu et al., 2024; Guo et al., 2024).

Detectors struggle with generalization due to linguistic and domain confounders, which introduce spurious correlations that bias detection models (Dai et al., 2022a; Voita et al., 2024). Linguistic confounders, such as sentence length and lexical diversity, reflect training data artifacts rather than intrinsic AI text features, leading to poor out-of-distribution (OOD) performance (Leng and Xiong, 2025; Wu et al., 2025b).

Domain confounders further complicate detection by linking AI-generated text with specific topics or styles rather than universal generation patterns (Wang et al., 2024b; Doughman et al., 2025). For instance, detectors trained on academic AI text often fail with news articles, revealing a lack of cross-domain robustness (Wang et al., 2024b).

We propose a neuron-level framework of training-based AI text detectors, leveraging the concept of **Confounding Neurons**: specific neuronal activations that encode dataset-specific biases (Pan et al., 2024; Voita et al., 2024). By systematically identifying these neurons, we can analyze how and where spurious correlations emerge in detectors, enabling the development of generalizable detection strategies that prioritize intrinsic textual features rather than dataset-dependent artifacts. We show an example of the effectiveness of our approach in Figure 1.

Emerging research indicates that LLMs encode knowledge, writing styles, and topic preferences within specific neurons (Dai et al., 2022b; Tang et al., 2024; Zhao et al., 2025). While prior work has examined neurons in the context of knowledge storage and language generation, their role in AI-generated text detection remains largely unexplored. Given that AI text detectors unintentionally encode dataset-specific biases, understanding their neuronal activations is crucial for disentangling the detection task from these confounding factors. We focus on computationally inexpensive and validated techniques within Knowledge Editing (Wang et al., 2024a) to build a framework for intervening at the neuronal level, enabling more reliable detection systems that generalize across text distributions.

Contributions This study advances robust LLM–generated text detection through the following contributions:

- Introduction of confounding neurons in the context of LLM-generated text detection.
- Development of an experimental framework for identifying and mitigating confounding neurons to improve detector performance.
- Analysis of neuron localization, showing that early-layer neurons can boost OOD accuracy while maintaining in-domain performance
- Evaluation of neuron-ranking methods, identifying critical neurons whose removal enhances generalization and accuracy.

2 Related Works

Detection methods are mainly categorized into Statistical methods, Neural-based and LLM-based detectors. Statistical methods detect AI-generated text by analyzing linguistic features such as perplexity, n-gram frequency, or token distribution (Hamed and Wu, 2024; Yang et al., 2024). These methods are computationally efficient and perform well for simple LLMs, but their effectiveness decreases when faced with larger more advanced models (Wu et al., 2025a). Neural-based detectors, employing transformer architectures like BERT (Devlin et al., 2019), RoBERTa (Zhuang et al., 2021), and XLM-R (Chi et al., 2022), achieve high accuracy (often exceeding 99%) on controlled datasets (Zeng et al., 2024). However, their performance degrades significantly on out-of-distribution (OOD) data, revealing limited generalization (Wu et al., 2025a).

A fundamental challenge for detection systems is achieving OOD robustness. Despite their high accuracy within specific domains, neural detectors struggle with diverse text types (Wang et al., 2024b), as linguistic and domain confounders introduce spurious correlations that hinder generalization across domain shifts (Wu et al., 2025a; Dai et al., 2022a).

Generalization in AI-Generated Text Detection

In this direction, Wang et al. (2024b) introduced the M4 benchmark, a large-scale dataset designed to evaluate detection models across multiple AI generators and linguistic styles. Their findings revealed that most models exhibit severe performance degradation when tested on OOD data.

Similarly, Lekkala et al. (2025) investigated domain-specific biases in AI text detection, demonstrating that models trained on one dataset struggle to adapt to new text domains. Gritsai et al. (2024) reinforced these findings by analyzing dataset quality issues, concluding that models are often trained on unrepresentative samples, leading to poor real-world adaptability. Wu et al. (2025b) benchmarked several detection techniques in real-world settings, revealing that even high-performing models struggle with cross-domain generalization. Gui et al. (2025) proposed AIDER, a robust topic-independent model that generalizes well across multiple domains using domain adaptation techniques.

Studies such as Fraser et al. (2025) and Dough-

man et al. (2025) have also shown that detection models perform poorly on short-form AI-generated content, such as news articles, where stylistic differences between AI- and human-generated text are less pronounced. Lee et al. (2024) demonstrated that reward-based learning techniques can improve robustness, but even these models fail when confronted with adversarially optimized text.

Neuron-Level Interpretability and Detection

A rapidly growing area in LLMs interpretability research have explored how neurons can store factual knowledge and respond to specific concepts and how we can exploit these findings to perform model interventions, that is, local modifications of a LM performed after training for improving efficiency, knowledge editing, or unlearning (Wang et al., 2024a). We can roughly categorize knowledge discovery in transformer-based models in activation-based (Voita et al., 2024), attribution-based (Dai et al., 2022b), and probing (Gurnee et al., 2023).

In Suau et al. (2024), neural intervention is used to reduce toxic outputs in text generation tasks, Tang et al. (2024) argue that a small subset of neurons is responsible for language selection in multilingual models. Chen et al. (2025) employ attribution-based methods for finding clusters of query-relevant neurons in LLMs for long-form texts, while Dai et al. (2022a) use gradient-based methods to trace neurons connected to syntactic phenomena and discuss the practical relevance of interventions on those neurons. To the best of our knowledge, this is the first study of confounding neurons in text detection systems.

3 Methods

The guiding hypothesis is that the cross-domain fragility of modern AI–text detectors originates in a *small, localized subset of neurons* whose activity encodes linguistic and domain confounders rather than generation source-specific signals. If these neurons are identified and "deactivated" after training in a *post-hoc* approach, the detector should preserve in-domain accuracy while exhibiting better generalization to unseen text distributions. To examine this hypothesis, we propose a model-agnostic framework (Figure 2) based on neuron-level intervention:

1. **Domain-aware data partitioning (§3.1):** construct a three-way split (train / in-domain / OOD) that effectively separates domains and topics, enabling controllable distribution shifts.

2. **Detector Training** (§3.2): fine-tune a pre-trained transformer (BERT in our running example) on the training split to obtain the reference model M_0.

3. **Confounding-neuron discovery (§3.3):** Identify neurons correlated with domain-specific cues by extracting topic-salient keywords, scoring hidden units for keyword sensitivity, and aggregating scores through a *label-stratified top-K intersection*.

4. **Neuron patching / model steering** (§3.4): mask the feed-forward layers of the transformer blocks at inference time to create a patched model M_p effectively removing confounders from the inference path.

The framework allows for a controlled comparison between M_0 and M_p on identical inputs; any gain in OOD performance can thus be attributed to the removal of the confounding neurons. The remainder of this section details each stage, and §4 reports the empirical findings.

3.1 Domain-Aware Data Partitioning

We frame LLM-generated text detection as a binary sequence-classification task over an open set of textual domains. Given a labelled corpus

$$\mathcal{D} = \{(t_i, y_i)\}_{i=1}^{N}, \quad y_i \in \{0 \text{ (human)}, 1 \text{ (LLM)}\},$$

the goal is to learn a detector $M_0 : t \mapsto [0, 1]$ that predicts the origin –human or machine– of texts, and exploit our framework to mitigate model degradation when applied *previously unseen* texts domains and topics. The degradation stems from spurious cues, including linguistic confounders (e.g., sentence length, lexical diversity) and domain confounders, where specific topics or styles are incorrectly linked to authorship (Doughman et al., 2025). This work primarily addresses the latter.

To disentangle genuine generative signals from these confounders, we impose a three-way partition that can be in principle instantiated on any multi-domain corpus.

Let J be the set of topics in domain A and choose a random subset $J_{\text{train}} \subset J$.

$$\mathcal{D}_{\text{train}} = \{(t, y) \mid \text{dom}(t) = A,\ \text{topic}(t) \in J_{\text{train}}\},$$
$$\mathcal{D}_{\text{OOS}} = \{(t, y) \mid \text{dom}(t) = A,\ \text{topic}(t) \in J \setminus J_{\text{train}}\},$$
$$\mathcal{D}_{\text{OOD}} = \{(t, y) \mid \text{dom}(t) = B,\ \text{topic}(t) \in J_{\text{OOD}}\},$$

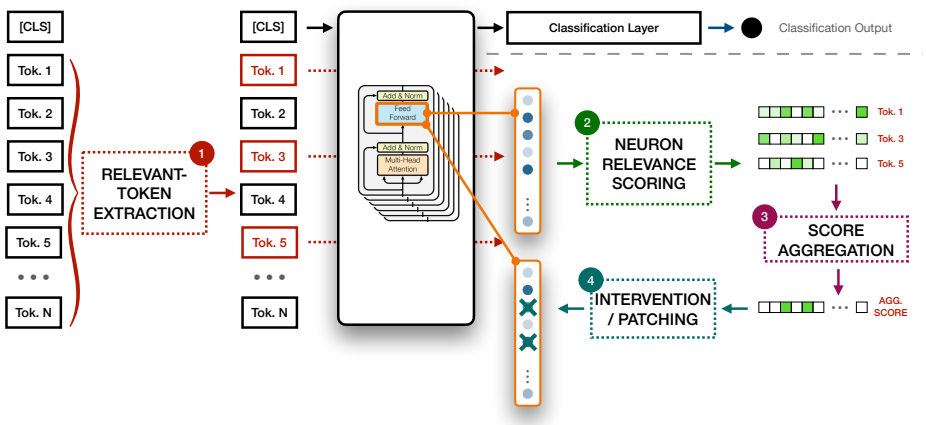

Figure 2: **Confounding Neuron** framework: Given a text corpus, for each topic we extract the most topic-related keywords ❶, and from the output embeddings of each keyword we compute the relevance score of the transformer MLPs' neurons for each text ❷. The scores are aggregated across texts and keywords obtaining a relevance score matrix for each neuron in each transformer layer ❸. Finally, the top-K neurons (Confounding Neuron) are suppressed based on the score ranking in order to improve the classification accuracy ❹.

Split	Domain(s)	Topic(s)
Train ($\mathcal{D}_{\text{train}}$)	A	J_{train}
OOS ($\mathcal{D}_{\text{OOS}}$)	A	$J_{\text{test}} = J \setminus J_{\text{train}}$
OOD ($\mathcal{D}_{\text{OOD}}$)	$B \neq A$	any J_{OOD}

Table 1: Domain-aware three-way dataset partition.

where $J \cap J_{\text{OOD}} = \varnothing$ and $A \neq B$. This partition controls both topic-level $\{J_{\text{train}}, J \setminus J_{\text{train}}, J_{\text{OOD}}\}$ and domain-level $\{A, B\}$ distribution gaps.

3.2 Detector

In this work we assume to have a detector trained for the task of binary LLM-vs-human text classification. In particular, we use a *pre-trained transformer encoder* that has been fine-tuned only on the training split $\mathcal{D}_{\text{train}}$. Unless otherwise specified, we take a BERT encoder followed by a fully-connected classification head, and refer to the resulting model as M_0. This choice reflects standard practice in recent studies of AI-text detection and provides a clear reference point for the neuron-level analysis that follows. The purpose of selecting a single baseline detector is purely expository: it allows us to trace how neuron-level interventions modify a specific network while clearly demonstrating that the framework can be applied to alternative detection models (e.g., RoBERTa, mBERT).

3.3 Confounding-Neuron Discovery

The aim of this stage is to pinpoint individual neurons whose activity tracks *confounding factors*, as topic, genre, length, surface style, rather than relevant generation cues (Voita et al., 2024; Pan et al., 2024). We decompose the procedure into three modular blocks that can be instantiated with either unsupervised or supervised topic information.

(i) Relevant-token extraction. The goal is to get a set $\mathcal{K} = \{k = 1, \ldots, K\}$ of topic-salient token indices that are *potentially* irrelevant to the detection.

Two alternative routes are available Supervised or Unsupervised, depending if topic labels are given in the considered corpus. For the supervised case, $\mathcal{D}_{\text{train}}$ is divided by topics J_{train} and labels $\{0, 1\}$. Within each slice, the top–K tokens are retrieved and ranked by TF-IDF (term frequency-inverse document frequency) weighting. In the unsupervised setting, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is applied to $\mathcal{D}_{\text{train}}$, retaining the top-$K$ tokens with the highest component probability from each latent components

(ii) Neuron-level relevance scoring. Considering a BERT-based detector, each transformer block contains an intermediate dense layer in its feed-forward network (FFN); we examine the H=3072 hidden units of each of its L=12 transformer layers, which are known to store factual, stylistic, and topic features (Dai et al., 2022b; Pan et al., 2024).

For each transformer block $\ell \in 1, \ldots, L$ and each neuron $h \in 1, \ldots, H$, let $a_{\ell h}(t_{ik})$ denote the activation of neuron (ℓ, h) (prior to applying the nonlinearity) for the token at position j in the i-th input text. The first is the Integrated Gradient (IG)

(Sundararajan et al., 2017). For each text in a sample $x_i \in \hat{\mathcal{D}}_{\text{train}} \subset \mathcal{D}_{\text{train}}$ and topic-related keyword index $k \in \mathcal{K}(x_i)$, the IG score is computed as:

$$w^{\text{IG}}_{i\ell hk} = \text{IG}\big(a_{\ell h}(t_{ik})\big),$$

The second is the topic-prediction Average Precision score (AP), inspired by (Suau et al., 2024). Given the train corpus $\mathcal{D}_{\text{train}}$, we build $|J_{train}|$ complementary labeled datasets $\mathcal{D}_j = \{(x_i, c_i)\}_{i=i}^{N},\ j \in J_{train}$, where $c_i = j$. For each neuron (ℓ, h) and keyword k of an input text x_i of topic c_i, the activation value $a_{\ell h}(t_{ik})$ is used as a one-vs-rest predictor of c_i, and the relevance score is then given by the Average Precision score (AP) over $\mathcal{D}_j$ with $j \in J_{train}$:

$$w^{\text{AP}}_{j\ell hk} = AP(\{(a_{\ell h}(t_{ik}), c_i), (x_i, c_i) \in \mathcal{D}_j\})$$

The tensor w is then reduced across texts and either the keywords dimension for IG or the topic dimension for AP, yielding an $L \times H$ importance matrix from which the confounding-neuron ordered sequence $\mathcal{C}$ is derived.

(iii) Score aggregation. Several aggregation strategies can map the w relevance tensor onto the $L \times H$ final neuron relevance representation, taking into account that we want to simultaneously minimize the importance of the selected neurons for the final text detection task. We adopt a label-stratified top-K intersection scheme that pinpoints neurons whose largest relevant score are driven by topic keywords in both classes, as a proxy for a purely spurious correlation. Namely, for each text label $y \in \{0, 1\}$ we take the maximum across all the extra dimensions (qualitatively similar results are obtained by taking the mean) and we keep the top-K' highest scoring indices, obtaining two ordered lists $R^{(0)}$ and $R^{(1)}$. The final relevance score matrix entries S_{lh} are obtained by taking the intersection $R^{(0)} \cap R^{(1)}$ while assigning the maximum scores between the two labels for each neuron, keeping the top-K with $K \leq K'$ indices, and setting all the other indices to zero.

Finally, from the matrix $\mathbf{S}$ we obtain the ordered confounding neurons sequence $\mathcal{C} = \langle(\ell, h) \mid S_{\ell h} \geq 0\rangle$ that highlights hidden units that consistently align with topic keywords across both author labels, making them prime candidates for the patching intervention in §3.4.

3.4 Neuron Patching and Model Steering

From the confounding neurons sequence $\mathcal{C}$, we intervene on the baseline detector M_0 without touching any other parameters. We define a binary mask $\mathbf{m}_\ell \in \{0, 1\}^H$ for each block such that $\mathbf{m}_\ell[h] = 1 \iff (\ell, h) \in \mathcal{C}$. The mask is frozen and applied at run time; no additional learning is performed. For every input text t and block ℓ, let $\mathbf{a}_\ell(t) \in \mathbb{R}^H$ be the activations of the *intermediate* feed-forward layer (see §3.3). We apply:

$$\tilde{\mathbf{a}}_\ell(t) = (1 - \mathbf{m}_\ell) \odot \mathbf{a}_\ell(t) + \mathbf{m}_\ell \odot g\big(\mathbf{a}_\ell(t)\big), \quad (1)$$

where $g(\cdot)$ is a patching policy. Several policies can be applied (Wang et al., 2024a; Voita et al., 2024; Pan et al., 2024), such as hard ablation $g(\mathbf{a}) = \mathbf{0}$, soft scaling $g(\mathbf{a}) = \alpha\mathbf{a},\ 0 < \alpha < 1$, or noise injection. In this work we consider the *hard ablation*, that is, the complete suppression of the considered neuron obtaining the final patched detector M_p.

4 Experiments

In this section, we evaluate the proposed neuron-level intervention framework for LLM-generated text detection[1]. The experiments are designed to assess the efficacy of our approach in addressing domain generalization challenges.

We aim to answer three research questions:

RQ1 Localisation & distribution: where are confounding signals concentrated, and how do they spread across layers?

Finding: Just 20 neurons in the early transformer blocks govern up to $+7\%$ accuracy gains on OOD text, while leaving in-domain performance intact, meanwhile, task-relevant "detection" neurons cluster almost exclusively in the final layers (Fig. 1, Fig. 4).

RQ2 Representation geometry: how does suppressing confounding neurons reshape the detector's embedding space?

Finding: Patching collapses topic-driven clusters in the classification embedding space, and increasing detector specificity on unseen domains (Fig. 3).

RQ3 Attribution robustness: do different neuron-ranking methods yield consistent improvements and similar high-leverage neurons?

Finding: Integrated Gradients exposes a handful of neurons whose removal causes stepwise accuracy jumps, whereas probing allows to

[1]Code available at https://anonymous.4open.science/r/confounding_neurons-1EC2

pruning up to $\sim 30\%$ of the FFN units with comparable OOD gains (Fig. 7).

4.1 Dataset description

To evaluate the proposed neuron-level intervention framework, we conduct experiments using three publicly available datasets commonly used in LLM-generated text detection: DAIGT[2], HC3 (Guo et al., 2023), and XSum from Li et al. (2024). These datasets differ significantly in text type, generation methods, and domain, allowing for a robust evaluation of out-of-distribution (OOD) generalization.

DAIGT is a large collection of student essays based on the Persuade corpus (Crossley et al., 2024), covering 23 different topics, generated using 11 different models. *HC3* is a Q&A dataset where responses are generated by ChatGPT. *XSum* is a news summarization dataset with texts generated using GPT-based models. We summarize the key statistics of each dataset in Table 2.

In all the experiments, we select a subset of topics/domains from a given dataset such as DAIGT or HC3 that constitute the training dataset $\mathcal{D}_{train}$ for training the base detector M_0. The split is performed with the Domain-Aware Data Partitioning §3.1. We note that even if the absolute performance of the detector is not important in our work, we always obtain a in-domain, in-sample test accuracy above 97%, in line with state-of-the-art models.

After training, the topics excluded from train on the same dataset constitute the OOS test set, and the remaining datasets constitute the OOD test sets.

4.2 Results

To evaluate the proposed framework, we conducted extensive experiments using multiple datasets §4.1 and various framework configurations §3. To address the main research questions, we present specific instantiations of the framework in the main text, while additional parameterizations and experimental variations are provided in the appendix §A.

Detection Generalization Improvement In Figure 1 we show an example of the efficacy of our method in identifying relevant confounding neurons. We fine-tune the detector on two topics of the DAIGT dataset ("car-free cities" and "distance learning") and use Integrated Gradients (IG) to identify the most important confounding neurons as described in §3. We then proceed to gradually remove the top-K neurons (here $K = 50$) one at the time and observe the difference in detection accuracy for three test sets, i.e., DAIGT samples from different topics (OOS), HC3 (OOD), and Xsum (OOD). We observe that removing as few as 20 neurons can bring an improvement in detection accuracy up to 7% in texts OOS and 3% in OOD, with specific single neurons responsible for sudden jumps of around 3% in OOS detection accuracy. The shaded area acts as a baseline and depicts the effect of randomly suppressing neurons from intermediate layers, showing minimal effects in the overall accuracy, as expected.

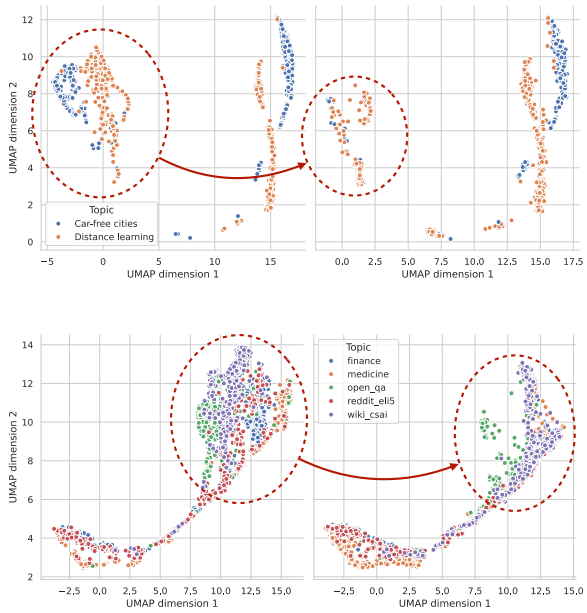

Figure 3: UMAP projection of the classification embedding space for M_0 (left panels) and M_p (right panels) on $\mathcal{D}_{train}$ (top) and $\mathcal{D}_{OOD}$ (bottom).

In Figure 3 we show a 2D representation of the output [CLS] embeddings, that is, the input to the final classification layer, obtained using UMAP for the DAIGT Train (top) and HC3 OOD test (bottom) datasets both without intervention (left panel) and after removing the top-2 confounding neurons found by our method (right panel). Each dot represents a textand each color represents a topic. For the top panel, human- and machine-generated texts are well separated, with machine-generated texts being almost perfectly clustered on the left (see Figure 11 in the Appendix). For the bottom panel, the labels are less well separated, as expected, with right-top lobe is most associated to machine-generated texts. As shown, the original model clusters well the topics of each text, but after

[2]https://www.kaggle.com/datasets/thedrcat/daigt-v4-train-dataset

Dataset Name	# Humans	# Machine	Confounders/Topics	Text Type	Generators
DAIGT	27,371	17,497	23	Student Essays	11
HC3	24,322	23,867	5	Q&A	ChatGPT
XSum	3,259	5,991	1	News Articles	GPT-based

Table 2: Summary of the considered human- and machine-generated text data.

the intervention on the confounding neurons the embeddings are collapsed and the detection model is not able to separate the topics anymore.

Distribution of Confounding Neurons An interesting aspect of confounding neurons in LLM-generated text detection models is their distribution across the transformer layers of the detector and the comparison with the intermediate neurons that are more associated with the main detection task. In Figure 4 we compare the distribution of the top-50 confounding neurons and the top-50 "detection" neurons obtained by applying our neuron attribution method to the classification token [CLS] instead of the topic keywords. While the detection neurons are concentrated almost exclusively at the final layers, confirming a general observation in mechanistic interpretability (Dai et al., 2022b; Bereska and Gavves, 2024). In contrast, confounding neurons are more prevalent in the initial layers, suggesting that the model processes topic-related concepts early on and then propagates this information to the later layers for final detection.

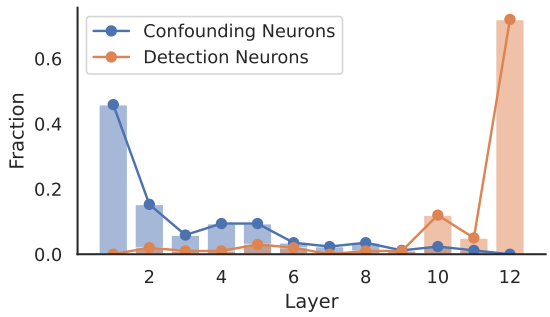

Figure 4: Distribution across layer of confounding neurons (using IG scoring) and detection-relevant neurons.

Supervised vs. Unsupervised topic definition Unsupervised topic modeling techniques are powerful and scalable but may introduce strong spurious correlations with the text detection task. As illustratory example, we consider the DAIGT dataset with known ground-truth topics and compute the LDA with total components $n_c = 2n_{\text{gt}}$ where n_{gt} is the number of ground-truth topics and 2 takes into account the binary classification task.

As shown in Figure 12 in the Appendix, there is a good correspondence between ground-truth topics and LDA components with many components including only one topic. In Figure 5, we show an illustrative example. Many components seem to separate very well the human and LLM-generated texts. While components 7 and 26 map to a single topic and present a balanced mix of the two classification labels (check Figure 12), components 15 and 24 separate perfectly the detection labels in the same ground-truth topic.

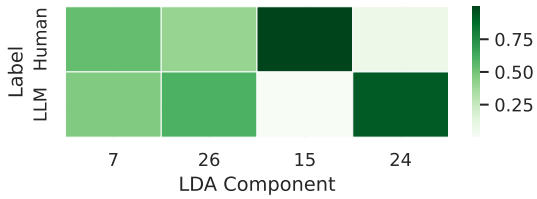

Figure 5: LDA topic components can encode detection label information, revealing intrinsic biases in human-vs-LLM generated detection datasets.

It is reasonable to assume that components such as 15 and 24 can be utilized to extract relevant neurons for the detection task. Consequently, suppressing these neurons would likely lead to a reduction in detection accuracy, as it becomes challenging to effectively separate topic-related confounding factors from detection related information. In contrast, components like 7 and 26, which map to individual topics and maintain label balance, are ideal candidates for identifying confounding neurons.

Figure 6 confirms this hypothesis: a model trained on components 15 and 24 (top) exhibits a decrease in detection accuracy when our framework is applied, whereas a model trained on components 7 and 26 (bottom) shows the opposite effect.

Comparison of Different Neuron-level Relevance Scoring.

We compare two different neuron-level relevance scoring methods, as described §3.3. The

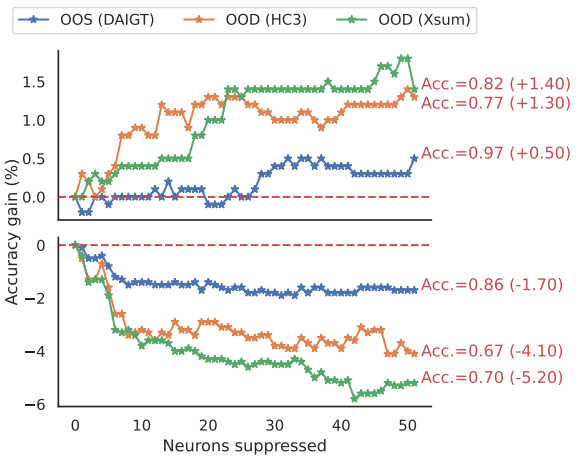

Figure 6: Relative variation in detection accuracy on the in-domain, out-of-sample (OOS), and two out-of-distribution (OOD) test sets of removing the top-K ($K = 1 - 50$) confounding neurons when the model is trained on LDA topics that are not disentangled from the task labels (top) and when the topics are not informative for the detection task (bottom).

Integrated Gradient IG-based scoring method is derive by Knowledge Neurons (Dai et al., 2022b), meanwhile the Average Precision insipred by the Expert Neurons (Suau et al., 2024). Both methods aim to identify neurons that encode specific knowledge directly related to the final task (e.g., text generation). Our framework, however, is designed to find confounding neurons, that capture spurious correlations rather than essential information, helping to improve the model's generalization.

A first comparison of IG- and AP-based relevant scoring can be appreciate in Figure 14 (for AP) and 4 (for IG): we obtain a very similar distribution of topic vs. detection neurons across layers.

A second comparison focuses on identifying confounding neurons. While the AP-based method yields high scores and demonstrates near-perfect topic classification capabilities (Figure 13), it is not as effective as IG in identifying specific confounding neurons according to our definition.

Interestingly though, as shown in Figure 7, the AP-based method allows for the removal of even 30% of the total intermediate layer neurons in the feed-forward networks of the transformer blocks not only without loosing detection accuracy, but even improving it up to almost 7% for the OOD datasets. This kind of phenomenology is not new, as it is known that transformer-based models for NLP tasks are extremely redundant (Dalvi et al., 2020), it is worth noting the striking differences in the two neuron ranking approaches: the IG-based attribution score is able to identify a few confounding neurons that correspond to sudden jumps in detection accuracy, while the AP-based score fail to recover these specific neurons but allows for an extreme pruning of the detector while reaching OOD detection accuracy that is comparable or even better than the IG-based approach.

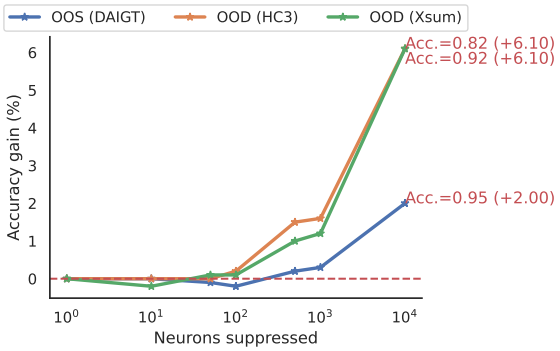

Figure 7: Effect on the in-domain, out-of-sample (OOS), and two out-of-distribution (OOD) test sets of removing the top-K ($K = 1-50$) confounding neurons computed using the AP method for detection task.

5 Conclusions

Through extensive experiments, we demonstrated that fine-tuning text detectors on specific domains or topics can lead to the emergence of confounding neurons: neurons that capture spurious correlations associated with concepts orthogonal to the detection task. These confounding neurons significantly compromise the model's ability to generalize to unseen domains and topics.

Our Framework shows that identifying and suppressing a small number of these confounding neurons within the intermediate layers of transformer-based models can effectively mitigate this issue, resulting in substantial improvements in out-of-distribution performance. The proposed method leverages simple yet effective neuron-relevance scoring techniques, such as gradient-based attribution and linear classification, without requiring any retraining, making it scalable to larger models.

While the current focus is on LLM-generated text detection, the proposed neuron-level intervention framework is general and can be applied to other text classification tasks where robustness to domain shifts is crucial. Future work will investigate extending this approach to a broader range of classification challenges.

Limitations

While our proposed framework effectively improves the generalization of LLM-generated text detectors, it also presents several limitations. The approach is primarily empirical, and we lack precise control over which confounding factors are being captured. This limits our ability to fully explain the differences observed between the two neuron-scoring methods (IG vs. AP) and to ensure coverage of all relevant confounding dimensions beyond topic and domain.

The datasets used for evaluation, although diverse, may not reflect the full complexity of real-world scenarios. For instance, the performances of LDA in separating labels suggest that existing benchmarks might be relatively "easy," lacking adversarial examples or deeper semantic variation. As a result, further evaluation on more challenging and diverse datasets is necessary to better assess robustness.

Our analysis also focuses exclusively on neurons in the feed-forward layers of the transformer architecture, omitting attention mechanisms and other components. While this already provide significant improvements, incorporating other layers and architectures could offer additional insights.

Moreover, all experiments are conducted using a single model architecture (BERT) for exposition clarity. Although this aligns with prior work and facilitates neuron-level analysis, confirming the generality of the framework across other architectures such as RoBERTa or mBERT is an important direction for future research.

Finally, the framework currently relies on keyword-based methods (LDA or supervised topic extraction) to localize confounding neurons. This assumes that spurious correlations are lexically grounded, which may not hold for more abstract or stylistic confounders. Developing alternative approaches that leverage higher internal representations of the detection model (i.e. sparse autoencoder SAE) could help uncover a broader range and more detailed confounding factors.

References

Alimohammad Beigi, Zhen Tan, Nivedh Mudiam, Canyu Chen, Kai Shu, and Huan Liu. 2024. Model attribution in llm-generated disinformation: A domain generalization approach with supervised contrastive learning. In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Lihu Chen, Adam Dejl, and Francesca Toni. 2025. Identifying query-relevant neurons in large language models for long-form texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23595–23604.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182, Dublin, Ireland. Association for Computational Linguistics.

S.A. Crossley, Y. Tian, P. Baffour, A. Franklin, M. Benner, and U. Boser. 2024. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Assessing Writing*, 61:100865.

Xavier Suau Cuadros, Luca Zappella, and Nicholas Apostoloff. 2022. Self-conditioning pre-trained language models. In *International Conference on Machine Learning*, pages 4455–4473. PMLR.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022a. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022b. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA,*

June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Jad Doughman, Osama Mohammed Afzal, Hawau Olamide Toyin, Shady Shehata, Preslav Nakov, and Zeerak Talat. 2025. Exploring the limitations of detecting machine-generated text. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4274–4281, Abu Dhabi, UAE. Association for Computational Linguistics.

Kathleen C. Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2025. Detecting ai-generated text: Factors influencing detectability with current methods. *J. Artif. Int. Res.*, 82.

G. Gritsai, A. Voznyuk, and A. Grabovoy. 2024. Are ai detectors good enough? a survey on quality of datasets with machine-generated texts. *Presented at Preventing and Detecting LLM Misinformation (PDLM) at AAAI 2025*.

J. Gui, B. Cui, X. Guo, K. Yu, and X. Wu. 2025. Aider: A robust and topic-independent framework for detecting ai-generated text. In *Proceedings of ACL*, pages 1–12.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, ZHUO ZHANG, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024. Biscope: AI-generated text detection by checking memorization of preceding tokens. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

Ahmed Abdeen Hamed and Xindong Wu. 2024. Detection of chatgpt fake science with the xfakesci learning algorithm. *Scientific Reports*, 14(1):16231.

Hyunseok Lee, Jihoon Tack, and Jinwoo Shin. 2024. Remodetect: Reward models recognize aligned llm's generations. In *Advances in Neural Information Processing Systems*, volume 37, pages 2886–2913. Curran Associates, Inc.

Sai Teja Lekkala, Annepaka Yadagiri, Mangadoddi Srikar Vardhan, and Partha Pakray. 2025. CNLP-NITS-PP at GenAI detection task 3: Cross-domain machine-generated text detection using DistilBERT techniques. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 334–339, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Yongqi Leng and Deyi Xiong. 2025. Towards understanding multi-task learning (generalization) of LLMs via detecting and exploring task-specific neurons. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2969–2987, Abu Dhabi, UAE. Association for Computational Linguistics.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Haowen Pan, Yixin Cao, Xiaozhi Wang, Xun Yang, and Meng Wang. 2024. Finding and editing multi-modal neurons in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1012–1037, Bangkok, Thailand. Association for Computational Linguistics.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. 2024. Whispering experts: neural interventions for toxicity mitigation in language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2024. Neurons in large language models: Dead, n-gram, positional. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1288–1301, Bangkok, Thailand. Association for Computational Linguistics.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024a. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3).

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and

multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025a. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2025b. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. In *NeurIPS 2024 Datasets and Benchmarks Track*.

B. Xu, R. Wang, L. Ping, C. Zhu, and X. Liu. 2024. Mat: Medical ai-generated text detection dataset from multi-models and multi-methods. In *Proceedings of IEEE AI Conference*.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. In *ICLR*.

Cong Zeng, Shengkun Tang, Xianjun Yang, Yuanzhou Chen, Yiyou Sun, zhiqiang xu, Yao Li, Haifeng Chen, Wei Cheng, and Dongkuan Xu. 2024. DLAD: Improving logits-based detector without logits from black-box LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

You Zhou and Jie Wang. 2024. Detecting ai-generated texts in cross-domains. In *Proceedings of the ACM Symposium on Document Engineering 2024*, DocEng '24, New York, NY, USA. Association for Computing Machinery.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

A.1 Dataset details

As described in the main text, we consider three datasets:

- **DAIGT:** A large collection of student essays covering 23 different topics, generated using 11 different models.
- **HC3:** A Q&A dataset where responses are generated by ChatGPT.
- **XSum:** A news summarization dataset with AI-generated texts generated using GPT-based models.

DAIGT is particularly fit for our study since it contains several labeled topics with limited overlap and clear subjects. It's a collection of 44868 essays, 27371 human and 17497 LLM-generated, from different transformer-based models. In Table 3 we report the different generators used and their text frequencies. In Table 4 we report the topics and their frequencies.

Generative Model	Count
mistral7binstruct_v2	2,421
chat_gpt_moth	2,421
llama2_chat	2,421
mistral7binstruct_v1	2,421
kingki19_palm	1,384
train_essays	1,378
llama_70b_v1	1,172
falcon_180b_v1	1,055
darragh_claude_v6	1,000
darragh_claude_v7	1,000
radek_500	500
NousResearch/Llama-2-7b-chat-hf	400
mistralai/Mistral-7B-Instruct-v0.1	400
cohere-command	350
palm-text-bison1	349
radekgpt4	200

Table 3: DAIGT Generative Models

HC3 is a corpus of 48185 texts, with 24320 human and 23865 generated by ChatGPT (GPT 3.5).

Xsum is a balanced dataset of 6000 news, 3000 human and 3000 generated using GPT-based models, and it does not contain topic labels.

A.2 Additional Experiments

To confirm the validity of our approach, we tested our framework in different training settings. First, we trained the model on DAIGT varying the number of training topics, and considered the case of 4

Topic	Count
Distance learning	5554
Seeking multiple opinions	5176
Car-free cities	4717
Does the electoral college work?	4434
Facial action coding system	3084
Mandatory extracurricular activities	3077
Summer projects	2701
Driverless cars	2250
Exploring Venus	2176
Cell phones at school	2119
Grades for extracurricular activities	2116
Community service	2092
"A Cowboy Who Rode the Waves"	1896
The Face on Mars	1893
Phones and driving	1583

Table 4: DAIGT Topics

Topic	Count
reddit eli5	33769
finance	7866
medicine	2493
open qa	2373
wiki csai	1684

Table 5: HC3 Topics

different topics instead of 2. Results are shown in Figure 8

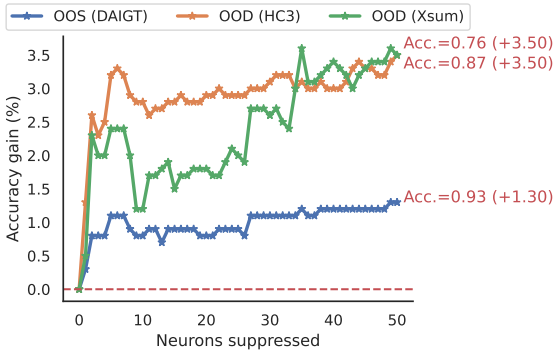

Figure 8: Accuracy gain in a human-vs-LLM detector BERT-based removing the top-K ($K = 1 - 50$) confounding neurons. DAIGT as training set with 4 different supervised topics.

In a different experiment, Figure 9, we train on 2 topics of HC3 and test on DAIGT and XSum as OOD test sets. We note that using HC3 as train set makes the extraction of confounding neurons more challenging since the labeled topics are actually different sources of human-generated texts, and consequently there may be more overlap across topics. Interestingly, the best improvement in detection accuracy is obtained for the OOS data, while the XSum OOD has always a very high accuracy score.

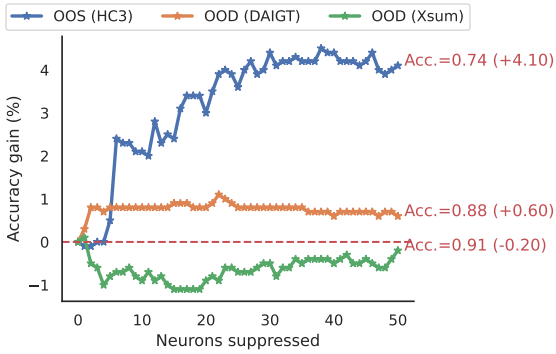

Figure 9: Accuracy gain in a human-vs-LLM detector BERT-based removing the top-K ($K = 1 - 50$) confounding neurons. HC3 as training set.

In table 6 we report a summary of the results for all the experiments, indicating the absolute values for accuracy in OOS and OOD (best of the two), both for M_0 and M_p. When topics are unsupervised and computed using LDA we denote with (LDA ↑) when it is expected to see an improvement in generalization and (LDA ↓) when we expect a reduction in detection accuracy because the extracted components do not allow a disentanglement of topic and label, as described in the main text.

Detection Neurons: In Figure 4 in the main text we discuss "detection" neurons, that is, neurons that are most related to the detection task and are well separated from the confounding neurons. Even if these neurons are in fact connected to the detection accuracy (see Figure 10), the focus of our study is to improve the OOD generalization by removing spurious data-related and domain biases and not acting on the neurons directly involved in the detection.

Finally, in Figure 11 we report the same embedding projection of Figure 3 color-coded for the text label.

A.3 Implementation and Training Details

We use the `transformers` python package with `torch` backend. The detector is a pretrained `bert-base-cased` sequence classification model, fine-tuned on the training dataset $\mathcal{D}_{train}$ for 4

Train dataset	# Train topics	Scoring	M_0 OOS	M_0 OOD	M_p OOS	M_p OOD	# Neurons
DAIGT	2	IG	0.92	0.76	**0.95**	**0.83**	50
DAIGT	4	IG	0.92	0.72	**0.93**	**0.76**	50
HC3	2	IG	0.69	0.87	**0.74**	**0.88**	50
DAIGT	2	AP	0.93	0.76	**0.95**	**0.82**	10000
DAIGT	2 (LDA ↑)	IG	0.96	0.75	**0.97**	**0.77**	50
DAIGT	2 (LDA ↓)	IG	**0.88**	**0.76**	0.86	0.70	50

Table 6: Summary of experiments.

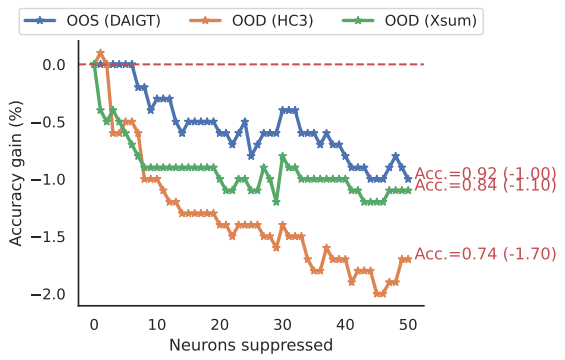

Figure 10: Accuracy variation in a human-vs-LLM detector BERT-based removing the top-K ($K = 1 - 50$) detection-relevant neurons.

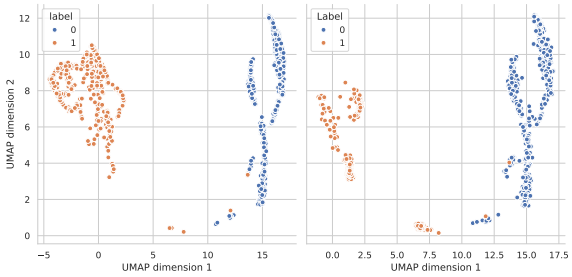

Figure 11: UMAP projection of the classification embedding space for M_0 (left panels) and M_p (right panels). Colors indicate human (blue) and LLM (red) generated texts.

epochs, with AdamW optimizer, learning rate lr $= 2 \cdot 10^{-5}$ and batch size bs $= 32$. Since BERT has a maximum positional encoding of 512, we consider up to the first 512 tokens for each text. In all experiments we consider 5 top keywords for each topic for topic relevance scoring.

A.4 Topics and Keywords Extraction

Supervised - TF-IDF: As detailed in the main text, in case of data with given ground-truth topics we extract the topic keywords by means of TF-IDF, that is, a well known statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (a corpus). It assigns a weight to each term in a document based on how frequently it appears in that specific document (Term Frequency) and how rare it is across all documents in the corpus (Inverse Document Frequency). TF-IDF gives a higher weight to terms that are frequent in a specific document but infrequent across the entire corpus. This helps to identify keywords that best characterize the content of a document.

Unsupervised - LDA: In case the considered training data do not have topic information, we employ Latent Dirichlet Allocation (LDA). LDA is a generative probabilistic model used for topic modeling. It is an unsupervised machine learning algorithm that aims to discover the underlying "topics" that occur in a collection of documents. LDA works by analyzing the co-occurrence of words within documents. It attempts to find groups of words that frequently appear together across different documents, inferring these groups as underlying latent topics. The model then determines the topic mixture for each document and the word distribution for each topic.

In Figure 12 we show the results when LDA to the whole DAIGT dataset as described in the main text and compared to the ground-truth topics.

A.5 Scoring Methods

Integrated Gradients. Integrated Gradients (IG) (Sundararajan et al., 2017) is an axiomatic attribution method used to explain the predictions of deep neural networks. It aims to determine the contribution of each input feature to the model's output. The core idea is to calculate the integral of the gradients of the model's output with respect to the input features, along a straight path from a baseline input to the actual input. The baseline is typically set to zero.

Mathematically, for an input x, a baseline x',

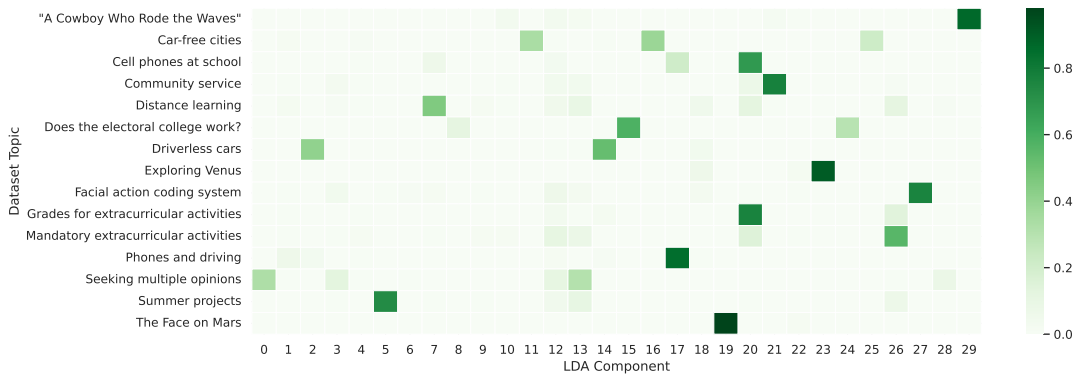

Figure 12: Distribution of the DAIGT corpus by LDA components and ground-truth topics. The number of components is given by # components $= 2 \cdot$ # ground-truth topics

and a model F, the attribution for the i-th feature x_i is defined as:

$$\text{IG}_i(x) \equiv (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial M(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

Here, α interpolates between the baseline and the input. Following Dai et al. (2022b), we consider a Riemann approximation of the Integrated Gradients where the integral is substituted by a discrete summation. In our experiments we found that already for 5 summation steps the results provide a good trade-off between computation cost and result accuracy.

Average Precision Following Cuadros et al. (2022), for each neuron we treat the activation value as the output prediction score of a linear classifier, and compute the Average Precision score (AP), that is, the area under the precision-recall curve using as output labels described in the main text. In Figure 13 we show the AP scores for each neuron when the detector is fine-tuned on DAIGT considering two topics. The surprising general high values of the AP score seem to indicate that almost half of all the MLPs neurons encode the ability to discriminate topics, in agreement with the results shown in the main text.

In Figure 14 we show the distribution of the most important 500 neurons using the AP score. In agreement with IG, most of the important confounding neurons are in the early layers of the detector, contrary to the detection neurons.

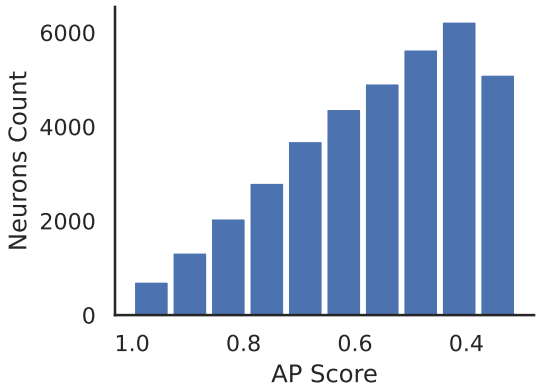

Figure 13: Distribution of the Average Precision scores for all MLP neurons across all layers.

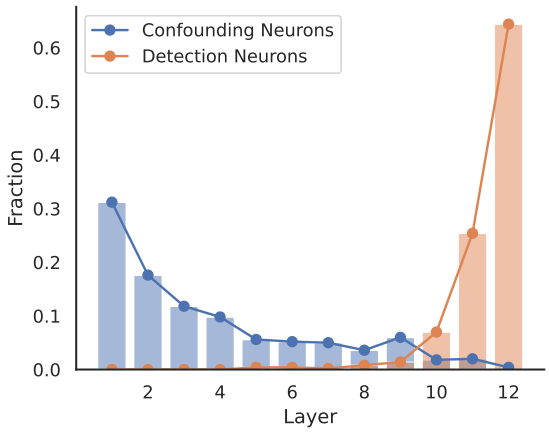

Figure 14: Distribution across layer of confounding neurons and detection-relevant neurons obtained using the AP score.