# LESIONPROBE: DETECTING AI-GENERATED TEXT THROUGH MODULE CONTRIBUTION PATTERNS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Detecting LLM-generated text requires signals that remain informative across generation sources and writing domains. Existing detectors primarily exploit textual features, predictive statistics, or hidden representations, without directly measuring how predictions depend on internal model components. We propose LesionProbe, which examines this dependence through component interventions in a frozen proxy LLM. It individually and jointly ablates attention and MLP branches and measures the resulting prediction shifts using normalized Jensen-Shannon divergence. These responses characterize module contribution patterns across layers and token positions, which a lightweight detector uses to distinguish human-written from generated text. Shapley analysis further reveals systematic layer-wise differences between the two text sources. On the DetectRL benchmark, LesionProbe maintains competitive in-distribution performance while achieving the best average detection results among the compared methods under both cross-source and leave-one-domain-out evaluation. It achieves 98.74% AUROC and 87.80% TPR at 0.1% FPR on unseen generation sources. On held-out writing domains, average AUROC reaches 98.92%, with a TPR of 87.66% at the same FPR. These results support module contribution patterns as transferable signals for AI-generated text detection.

## 1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities in language understanding and generation (Achiam et al., 2023; Guo et al., 2025), supporting applications in education (Ye et al., 2025), academic research (Turcan et al., 2026), and software development (Ran et al., 2026). Their increasing use in AI agents for social media simulation and interaction further expands the presence of generated content in online environments (Xue et al., 2026). As LLM-generated text (LGT) becomes harder to distinguish from human-written text (HWT), reliable detection is increasingly important for assessing textual provenance (Chen et al., 2025).

Existing methods distinguish LGT from HWT using various detection signals, including learned textual features, statistics of model predictions, and responses to text perturbation or rewriting (Wu et al., 2025). More recently, methods such as RepreGuard have extended detection to the hidden representations of proxy LLMs (Chen et al., 2025), showing that their internal states also contain useful information for distinguishing text sources. Despite these advances, detection performance can deteriorate across generation sources and writing domains, particularly when a low false-positive rate is required (Chen et al., 2025). These challenges motivate further investigation into the evidence available within a proxy model. Beyond the states produced when processing a text, we examine how the model's predictions depend on the internal components that produce them.

**Key Insight**. We explore whether the influence of a proxy LLM's internal components provides useful evidence for text detection. Specifically, *when the same model processes HWT and LGT, does removing a component change its predictions to different extents?* This question complements detection based on model predictions or hidden representations by examining how those predictions depend on particular components. We investigate both whether these responses distinguish HWT from LGT and whether their detection value transfers across generation sources and writing domains. We focus on the attention and MLP branches within each Transformer layer. Prior mechanistic interpretability studies have identified differentiated roles for these components across layers



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and tasks (Elhage et al., 2021; Stolfo et al., 2023; Ferrando et al., 2024), motivating their use as units of analysis. Inspired by lesion studies in neuroscience (Rorden & Karnath, 2004), we measure their influence through the changes in next-token predictive distributions caused by disabling their outputs. These changes characterize the model's sensitivity to component removal. We refer to their variation across layers and token positions as *module contribution patterns*.

Building on this perspective, we propose LesionProbe, illustrated in Figure 1. It ablates attention and MLP branches individually and jointly in a frozen proxy LLM and uses the resulting response patterns to train a lightweight detector. Joint ablations complement individual ablations by measuring the effect of removing both branches together. For analysis, we use Shapley attribution to account for how each branch's influence depends on the presence of the other. Comparisons on matched HWT-LGT pairs reveal systematic layer-wise attribution differences across generation sources, with the largest shifts concentrated in early and final layers. We evaluate LesionProbe on DetectRL (Wu et al., 2024) across generation sources, writing domains, proxy models, text lengths, and text transformations. Across the 12 settings with unseen generation sources, it achieves an average AUROC of $98.74\%$ and a true-positive rate of $87.80\%$ at a false-positive rate of $0.1\%$ (TPR@0.1%FPR). Leave-one-domain-out experiments further demonstrate generalization to unseen writing domains. In a separate mixed-source evaluation, ten proxy LLMs yield mean AUROC values ranging from $99.40\%$ to $99.91\%$, supporting the applicability of the approach across different proxy models. Additional experiments show competitive performance on short texts and under DIPPER paraphrasing, with a small performance decrease under text perturbations. Sparse-layer experiments further characterize the trade-off between feature extraction cost and detection performance. Our main contributions are as follows:

- We investigate how a frozen proxy LLM's predictions depend on its internal computations when processing HWT and LGT. Shapley-based analysis reveals systematic layer-wise differences in attention and MLP contributions, motivating their use as detection signals.
- We propose LESIONPROBE, which constructs module contribution tensors from prediction changes induced by individual and joint branch ablations. A lightweight detector combines sequential and statistical features from these tensors without fine-tuning the proxy LLM.
- We demonstrate strong detection performance and generalization across generation sources and writing domains, and examine sensitivity to proxy choice, text length, and text transformations. We further evaluate the computation-performance trade-off of sparse-layer interventions.

## 2 RELATED WORK

### 2.1 AI-GENERATED TEXT DETECTION

We focus on detecting LGT from existing samples without modifying the generation process. Prior work exploits several complementary sources of evidence, including textual features, predictive statistics, responses to input transformations, and internal model representations (Wu et al., 2025).

**Textual Features**. A common approach learns to distinguish HWT from LGT by fine-tuning a pretrained language model for text-source classification (Zellers et al., 2019; Guo et al., 2023). Such detectors learn discriminative features directly from labeled text. Their detection evidence is encoded in the representations learned for the classification task.

**Predictive Statistics**. Another line of work characterizes text through probabilities assigned by language models. Early signals include token log-likelihood, entropy, and rank (Gehrmann et al., 2019). Subsequent methods construct more informative scores from these quantities. DetectLLM introduces the Log-Likelihood Log-Rank Ratio (LRR) (Su et al., 2023), while Binoculars compares perplexity with cross-perplexity computed using two related language models (Hans et al., 2024). Fast-DetectGPT uses conditional probability curvature to capture differences in token selection between HWT and LGT (Bao et al., 2024). Predictive statistics can also support learned detectors: Ghostbuster constructs and selects features from token probabilities assigned by weaker language models, then trains a classifier on these features (Verma et al., 2024).

**Responses to Input Transformations**. These methods derive detection evidence by perturbing, rewriting, or regenerating the input text. DetectGPT (Mitchell et al., 2023) compares the log-likelihood of a text with those of its perturbed variants to estimate local curvature of the model's

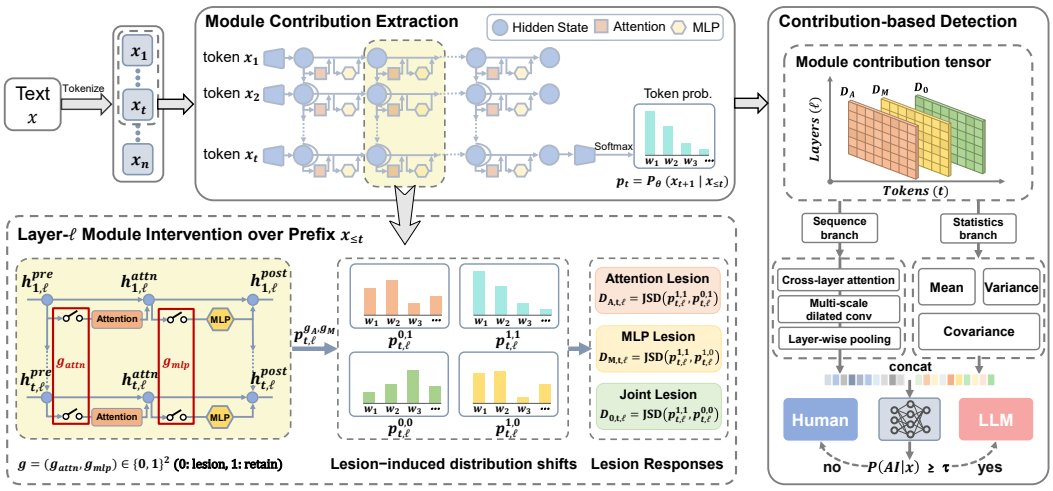


Figure 1: **Overview of LesionProbe.** Individual and joint attention/MLP ablations in a frozen proxy LLM yield prediction shifts measured by normalized JSD. The resulting module contribution patterns across layers and tokens enable a lightweight detector to distinguish LGT from HWT.

log-probability function. RAIDAR (Mao et al., 2024) uses differences between an input and its LLM-generated rewrites as detection features. Both approaches characterize text through its response to input transformations, measured through model scores or changes in textual content.

**Internal Representations**. More recently, detection methods have examined evidence within proxy LLMs. RepreGuard exploits source-related patterns in hidden representations to distinguish HWT from LGT (Chen et al., 2025). This approach demonstrates that internal states produced during text processing provide useful detection signals and motivates further investigation of the evidence available within a proxy model.

LesionProbe extends this investigation to the dependence of predictions on internal components. It keeps the input text fixed and measures changes in predictive distributions caused by removing attention and MLP branches. Its detection signal therefore comes from responses to model interventions. This perspective complements both input-transformation methods and internal-representation methods by examining how particular components contribute to the predictions made for a text.

### 2.2 Mechanistic Interpretability and Component Interventions

Mechanistic interpretability investigates how internal model components contribute to model behavior (Elhage et al., 2021; Ferrando et al., 2024). Studies of Transformer language models have identified differentiated and interacting roles for attention and MLP components. Geva et al. (2021) characterize feed-forward layers as key-value memories that associate input patterns with distributions over output tokens. Through activation interventions, Stolfo et al. (2023) trace arithmetic information transmitted by attention and subsequently processed by MLP components. In factual recall, Geva et al. (2023) identify computations involving subject-information enrichment by early MLP layers and attribute extraction through attention. These findings motivate using attention and MLP branches as units for examining input-dependent contributions to model predictions.

Interventions provide a means of assessing component influence through changes in model behavior. Following the general logic of lesion studies (Rorden & Karnath, 2004), LesionProbe measures this influence by disabling branch outputs and observing the resulting prediction changes. Individual and joint ablations characterize responses to removing either branch or both, while Shapley attribution supports an analysis of each branch's contribution across the presence and absence of the other.

Internal computation has also been used for diagnosis. For example, Sun et al. (2025b) use divergence between vocabulary-projected late-layer logits to characterize reasoning dynamics and detect reasoning hallucinations. LesionProbe investigates component-intervention responses as evidence for text-source detection. Its attribution analysis characterizes how the same frozen proxy model depends on its attention and MLP branches when processing HWT and LGT.

# 3 METHODOLOGY: LESIONPROBE

LesionProbe detects LGT through changes in a frozen proxy model's predictions under component removal, as illustrated in Figure 1. It independently ablates the attention and MLP branches at each Transformer layer and measures the resulting shifts in token-level predictive distributions. These responses form a module contribution tensor used to train a lightweight detector. We further analyze the responses using Shapley attribution to examine layer-wise differences between HWT and LGT.

## 3.1 PROBLEM FORMULATION

Given a tokenized text $x = (x_1, \cdots, x_n)$, we predict its source label $y \in \{0, 1\}$, where 1 denotes LGT and 0 denotes HWT. We use a frozen language model $f_\theta$ with $L$ Transformer layers as the proxy model. Only the detector parameters are optimized; the proxy model remains fixed throughout feature extraction and detector training. Details of the input formulation are provided in Appendix A. We characterize module contributions through changes in predictive distributions caused by disabling attention and MLP branch outputs. These changes measure the sensitivity of the proxy model's predictions to component removal for a given text. We refer to their variation across layers and token positions as *module contribution patterns*.

## 3.2 MODULE CONTRIBUTION EXTRACTION

We first obtain the intact model's next-token predictive distribution at each input position: $\mathbf{p}_t = p_\theta(\cdot \mid x_{\leq t})$, for $t = 1, \cdots, n$. A single forward pass with causal masking computes these distributions over the vocabulary. They serve as references for subsequent interventions.

**Component Lesions**. Following the logic of lesion studies (Rorden & Karnath, 2004), we assess component influence by disabling branch outputs and observing the resulting prediction changes. Each intervention affects one layer and applies to all token positions; downstream states are recomputed with all other layers unchanged.

Let $\mathbf{H}_{\ell-1} \in \mathbb{R}^{n \times d}$ denote the input hidden states to layer $\ell$, where $d$ is the hidden dimension. For a sequential pre-normalized Transformer layer, we introduce binary gates $g_A, g_M \in \{0, 1\}$ for the attention and MLP branches:

$$\begin{cases} \widetilde{\mathbf{H}}_\ell^{g_A} & = \mathbf{H}_{\ell-1} + g_A A_\ell \left(\mathcal{N}_{A,\ell}(\mathbf{H}_{\ell-1})\right), \\ \mathbf{H}_\ell^{g_A, g_M} & = \widetilde{\mathbf{H}}_\ell^{g_A} + g_M M_\ell \left(\mathcal{N}_{M,\ell}(\widetilde{\mathbf{H}}_\ell^{g_A})\right), \end{cases} \tag{1}$$

where $A_\ell$ and $M_\ell$ are the attention and MLP transformations, and $\mathcal{N}_{A,\ell}$ and $\mathcal{N}_{M,\ell}$ are their respective normalization operations. Setting a gate to zero disables the corresponding branch output while preserving the residual connection. When attention is disabled and the MLP remains active, the MLP is evaluated on the modified intermediate hidden states. We denote the resulting next-token distribution at position $t$ by $\mathbf{p}_{\ell,t}^{g_A, g_M}$. The configurations $(0, 1)$, $(1, 0)$, and $(0, 0)$ correspond to attention, MLP, and joint lesions, respectively. The intact configuration satisfies $\mathbf{p}_{\ell,t}^{1,1} = \mathbf{p}_t$.

**Lesion Responses**. We measure prediction changes using normalized Jensen-Shannon divergence (JSD)(Lin, 1991). For two distributions $\mathbf{p}$ and $\mathbf{q}$, let $\mathbf{m} = (\mathbf{p} + \mathbf{q})/2$. We define

$$\mathrm{JSD}(\mathbf{p}, \mathbf{q}) = \frac{D_{\mathrm{KL}}(\mathbf{p} \| \mathbf{m}) + D_{\mathrm{KL}}(\mathbf{q} \| \mathbf{m})}{2 \ln 2}, \tag{2}$$

where $D_{\mathrm{KL}}$ uses natural logarithms. This normalization yields a symmetric divergence in $[0, 1]$. For each layer and token position, we define the lesion response for configure $c$ as

$$D_{c,\ell,t} = \mathrm{JSD}\left(\mathbf{p}_t, \mathbf{p}_{\ell,t}^{\mathbf{g}_c}\right), \qquad c \in \{A, M, 0\}, \tag{3}$$

where $\mathbf{g}_A = (0, 1)$, $\mathbf{g}_M = (1, 0)$, and $\mathbf{g}_0 = (0, 0)$ correspond to attention, MLP, and joint lesions, respectively. These responses quantify changes in the final predictive distribution after the intervention has propagated through the model. The joint response complements the individual responses because removing both branches need not produce a divergence equal to the sum of their responses.

We stack the three responses into a module contribution tensor $\mathbf{D}(x) \in [0, 1]^{L \times n \times 3}$:

$$\mathbf{D}(x)_{\ell,t,:} = [D_{A,\ell,t}, D_{M,\ell,t}, D_{0,\ell,t}]. \tag{4}$$

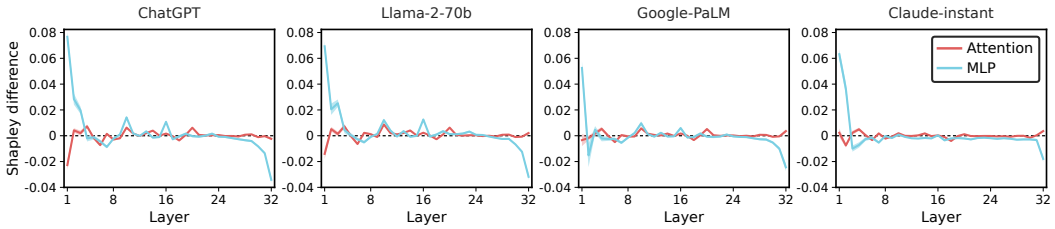


Figure 2: **Layer-wise Shapley differences between LGT and HWT.** Curves show mean paired differences (LGT minus HWT) for attention and MLP in frozen Llama-3.1-8B, using 2,800 matched pairs per source. Shading indicates pointwise 95% paired-bootstrap confidence intervals.

The tensor preserves variation across layers, token positions, and lesion types. A direct implementation requires $1+3L$ full-sequence forward passes: one shared intact pass and three lesion passes per layer. each producing responses for all token positions. Restricting extraction to $K$ selected layers reduces this count to $1+3K$.

### 3.3 Detection from Lesion Responses

We train a lightweight detector on the module contribution tensors. Its dual-branch encoder combines sequential patterns with statistical summaries. The sequence branch uses cross-layer attention and local convolutions over token positions to obtain a representation $\mathbf{g}(x)$. The statistics branch summarizes layer-wise channel means, variances, and covariances into $\mathbf{u}(x)$. A classification head takes their concatenation $[\mathbf{g}(x); \mathbf{u}(x)]$ and outputs the predicted LGT probability $q_\psi(x)$, where $\psi$ denotes the detector parameters. We classify $x$ as LGT when $q_\psi(x) > \tau$ and as HWT otherwise.

We optimize the detector using focal loss to reduce the weight of easily classified examples. For a batch of $B$ labeled texts, let $s_i = y_i q_\psi(x_i) + (1-y_i)(1-q_\psi(x_i))$ be the probability assigned to the correct class. The training objective is

$$\mathcal{L}_{\text{focal}} = -\frac{1}{B}\sum\nolimits_{i=1}^{B}(1-s_i)^\gamma \log s_i, \tag{5}$$

with $\gamma = 2$. Ablations of the feature branches, response channels, and training objectives are provided in Appendix D.

### 3.4 Shapley-based Contribution Analysis

The detector uses the three lesion responses directly. For analysis, we compute Shapley attributions to examine how attention and MLP contributions differ between HWT and LGT. A branch's influence may depend on whether the other branch is present. Shapley attribution averages its marginal contribution across these two contexts and allocates the joint lesion response between the branches.

**Coalition Utility**. For each layer $\ell$ and token position $t$, we define a two-player cooperative game with players $\mathcal{P} = \{A, M\}$. A coalition $S \subseteq \mathcal{P}$ specifies the branches retained at layer $\ell$, and $\mathbf{p}^S_{\ell,t}$ denotes the resulting predictive distribution. The empty coalition removes both branches, while the full coalition retains both.

Taking the joint-lesion configuration as the baseline and the intact prediction as the reference, we define

$$u_{\ell,t}(S) = D_{0,\ell,t} - \text{JSD}\left(\mathbf{p}_t, \mathbf{p}^S_{\ell,t}\right). \tag{6}$$

This utility measures the reduction in divergence from the intact prediction when coalition $S$ is retained. It satisfies $u_{\ell,t}(\varnothing) = 0$ and $u_{\ell,t}(\mathcal{P}) = D_{0,\ell,t}$.

**Exact Shapley Attribution**. With two players, the Shapley attribution (Kuhn & Tucker, 1953) for branch $k \in \mathcal{P}$ averages its marginal utility when the other branch is absent and present:

$$\Phi_{k,\ell,t} = \frac{1}{2}\sum_{S \subseteq \mathcal{P}\setminus\{k\}} \left[u_{\ell,t}(S \cup \{k\}) - u_{\ell,t}(S)\right]. \tag{7}$$

Substituting the lesion responses gives

$$\begin{aligned}\Phi_{A,\ell,t} &= \frac{1}{2}\left(D_{0,\ell,t} - D_{M,\ell,t} + D_{A,\ell,t}\right), \\ \Phi_{M,\ell,t} &= \frac{1}{2}\left(D_{0,\ell,t} - D_{A,\ell,t} + D_{M,\ell,t}\right).\end{aligned} \tag{8}$$

These attributions satisfy $\Phi_{A,\ell,t} + \Phi_{M,\ell,t} = D_{0,\ell,t}$, providing an additive allocation of the joint lesion response under the defined utility. They are computed exactly from the existing responses and require no additional model evaluations. An attribution can be negative because retaining an additional branch need not monotonically reduce divergence from the intact prediction.

**Text-level Contribution Profiles**. We average token-level attributions to obtain a layer-wise profile for each text:

$$\overline{\Phi}_{k,\ell}(x) = \frac{1}{n}\sum_{t=1}^{n} \Phi_{k,\ell,t}(x), \qquad k \in \mathcal{P}. \tag{9}$$

Given $N$ semantically matched pairs $(x_i^{\mathrm{LGT}}, x_i^{\mathrm{HWT}})$, we summarize source-related differences as

$$\Delta_{k,\ell} = \frac{1}{N}\sum_{i=1}^{N}\left[\overline{\Phi}_{k,\ell}(x_i^{\mathrm{LGT}}) - \overline{\Phi}_{k,\ell}(x_i^{\mathrm{HWT}})\right]. \tag{10}$$

Positive values indicate higher mean attribution for LGT and negative values for HWT. As shown in Figure 2, comparisons between matched HWT and LGT texts reveal systematic layer-wise attribution differences across generation sources, with more pronounced shifts in early and final layers.

All attributions are computed from the original JSD responses before detector preprocessing. Each layer defines a separate cooperative game, so the resulting profiles describe layer-wise contributions to prediction recovery rather than a whole-network Shapley decomposition. They characterize the proxy model's predictive dependence on its components when processing the two text sources.

## 4 Experiments

Our experiments address four research questions: **RQ1:** How effectively do module contribution patterns support detection and generalize across generation sources, writing domains, and benchmarks? **RQ2:** How do the lesion responses and detector components contribute to performance? **RQ3:** How sensitive is detection to the proxy model, text length, and text transformations? **RQ4:** What computation–performance trade-off results from restricting interventions to selected layers?

### 4.1 Experimental Setup

**Datasets**. We use DetectRL (Wu et al., 2024) for the main evaluations. It provides paired HWT and LGT from four generation sources: ChatGPT (OpenAI, 2022), Claude-instant (Anthropic, 2023), Google-PaLM (Google, 2023), and Llama-2-70B (Touvron et al., 2023). The data span arXiv abstracts, XSum news articles (Narayan et al., 2018), Writing Prompts stories (Fan et al., 2018), and Yelp reviews (Zhang et al., 2015). Dataset statistics are provided in Appendix B.1. We additionally evaluate cross-benchmark transfer using selected English subsets of RAID, MAGE, M4GT-Bench, AIGTBench, and DetectRL, as detailed in Appendix C.

**Baselines**. We compare LesionProbe with eight representative detectors: Likelihood (Gehrmann et al., 2019), LRR (Su et al., 2023), FastDetect-GPT (Bao et al., 2024), Binoculars (Hans et al., 2024), RoBERTa (Liu et al., 2019), RAIDAR (Mao et al., 2024), RepreGuard (Chen et al., 2025), and LTD (Zhou et al., 2026). Implementation details are provided in Appendix B.2. For cross-benchmark evaluation, we compare against LTD, RepreGuard, and RAIDAR.

**Metrics**. We report AUROC to measure overall discrimination and TPR at a false-positive rate of $0.1\%$ (TPR@0.1%FPR) to assess detection under a stringent false-positive constraint (Hans et al., 2024). The latter is particularly relevant when false identification of human writing is costly. Unless otherwise specified, TPR denotes TPR@0.1%FPR. Appendix E examines sensitivity to test-set size and reports confidence intervals for the low-FPR evaluation. The cross-benchmark experiments additionally assess conventional classification performance using Accuracy and F1 alongside AUROC, with thresholds fixed across target benchmarks as described in Appendix C.

Table 1: **Cross-source detection performance.** Reference source specifies fitting/calibration data ("None": unused). For named sources, matching and differing test sources indicate ID and OOD, respectively. Values are $\mathrm{mean}_{\pm\mathrm{std}}$ (%) over five runs where applicable. TPR denotes TPR@0.1%FPR; Average includes all four test sources. Blue/bold and gray mark the best and second-best means per named-source block; displayed ties share highlighting.

| Detector | Test Source | | | | | | | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ChatGPT | | Llama-2-70B | | Google-PaLM | | Claude-instant | | | |
| | AUROC | TPR | AUROC | TPR | AUROC | TPR | AUROC | TPR | AUROC | TPR |
| *Reference source: None* | | | | | | | | | | |
| Likelihood | 97.51 | 22.10 | 98.36 | 62.20 | 91.51 | 18.90 | 67.95 | 1.30 | 88.83 | 26.13 |
| LRR | 97.72 | 40.00 | 99.38 | 87.60 | 90.00 | 51.50 | 68.68 | 0.30 | 88.95 | 44.85 |
| FastDetect-GPT | 98.55 | 65.40 | 99.84 | 92.10 | 97.47 | 80.70 | 53.34 | 0.10 | 87.30 | 59.58 |
| Binoculars | 99.73 | 92.80 | 99.99 | 99.50 | 99.78 | 93.80 | 80.88 | 3.70 | 95.10 | 72.45 |
| *Reference source: ChatGPT* | | | | | | | | | | |
| RoBERTa | $99.78_{\pm 0.04}$ | $88.48_{\pm 7.07}$ | $98.88_{\pm 0.80}$ | $63.30_{\pm 9.50}$ | $95.27_{\pm 1.16}$ | $39.56_{\pm 9.88}$ | $\mathbf{96.81}_{\pm 0.76}$ | $24.78_{\pm 8.58}$ | $97.68_{\pm 0.60}$ | $54.03_{\pm 4.40}$ |
| RAIDAR | $87.20_{\pm 0.02}$ | $16.64_{\pm 1.70}$ | $87.67_{\pm 0.09}$ | $8.38_{\pm 0.80}$ | $69.74_{\pm 0.42}$ | $6.68_{\pm 0.58}$ | $69.78_{\pm 0.29}$ | $0.92_{\pm 0.31}$ | $78.60_{\pm 0.06}$ | $8.16_{\pm 0.63}$ |
| RepreGuard | $99.62_{\pm 0.10}$ | $99.38_{\pm 0.13}$ | $99.53_{\pm 0.08}$ | $99.40_{\pm 0.00}$ | $88.42_{\pm 0.49}$ | $76.22_{\pm 0.28}$ | $84.06_{\pm 1.33}$ | $66.28_{\pm 1.12}$ | $92.91_{\pm 0.45}$ | $85.32_{\pm 0.31}$ |
| LTD | $\mathbf{99.99}_{\pm 0.01}$ | $99.62_{\pm 0.74}$ | $98.03_{\pm 0.98}$ | $75.44_{\pm 10.80}$ | $92.35_{\pm 1.90}$ | $60.30_{\pm 9.97}$ | $84.12_{\pm 7.20}$ | $10.30_{\pm 5.13}$ | $93.62_{\pm 2.35}$ | $61.42_{\pm 5.66}$ |
| **LesionProbe** | $99.95_{\pm 0.04}$ | $\mathbf{99.86}_{\pm 0.05}$ | $\mathbf{99.99}_{\pm 0.01}$ | $\mathbf{99.60}_{\pm 0.29}$ | $\mathbf{98.81}_{\pm 0.26}$ | $\mathbf{87.20}_{\pm 1.52}$ | $95.66_{\pm 0.70}$ | $\mathbf{67.36}_{\pm 2.56}$ | $\mathbf{98.60}_{\pm 0.15}$ | $\mathbf{88.51}_{\pm 0.57}$ |
| *Reference source: Llama-2-70B* | | | | | | | | | | |
| RoBERTa | $99.51_{\pm 0.30}$ | $79.64_{\pm 9.19}$ | $99.57_{\pm 0.48}$ | $86.94_{\pm 10.44}$ | $96.46_{\pm 1.96}$ | $66.08_{\pm 7.15}$ | $96.83_{\pm 1.76}$ | $42.22_{\pm 14.08}$ | $98.09_{\pm 1.11}$ | $68.72_{\pm 8.10}$ |
| RAIDAR | $87.21_{\pm 0.01}$ | $18.22_{\pm 1.43}$ | $88.04_{\pm 0.06}$ | $10.26_{\pm 0.70}$ | $70.31_{\pm 0.32}$ | $7.04_{\pm 0.78}$ | $69.11_{\pm 0.25}$ | $1.12_{\pm 0.19}$ | $78.66_{\pm 0.03}$ | $9.16_{\pm 0.38}$ |
| RepreGuard | $99.51_{\pm 0.11}$ | $99.38_{\pm 0.04}$ | $99.49_{\pm 0.07}$ | $99.08_{\pm 0.04}$ | $89.61_{\pm 0.60}$ | $78.30_{\pm 0.56}$ | $85.85_{\pm 1.56}$ | $68.90_{\pm 0.72}$ | $93.62_{\pm 0.52}$ | $86.42_{\pm 0.12}$ |
| LTD | $\mathbf{99.99}_{\pm 0.00}$ | $\mathbf{99.90}_{\pm 0.00}$ | $99.98_{\pm 0.03}$ | $98.58_{\pm 1.83}$ | $98.66_{\pm 0.18}$ | $86.38_{\pm 5.72}$ | $\mathbf{99.70}_{\pm 0.30}$ | $\mathbf{89.20}_{\pm 5.66}$ | $\mathbf{99.58}_{\pm 0.12}$ | $\mathbf{93.52}_{\pm 2.08}$ |
| **LesionProbe** | $99.96_{\pm 0.03}$ | $99.86_{\pm 0.09}$ | $\mathbf{99.99}_{\pm 0.00}$ | $\mathbf{99.74}_{\pm 0.05}$ | $\mathbf{98.98}_{\pm 0.26}$ | $\mathbf{89.24}_{\pm 0.93}$ | $96.64_{\pm 0.79}$ | $69.76_{\pm 4.54}$ | $98.89_{\pm 0.21}$ | $89.65_{\pm 0.89}$ |
| *Reference source: Google-PaLM* | | | | | | | | | | |
| RoBERTa | $98.87_{\pm 1.26}$ | $60.12_{\pm 22.62}$ | $99.01_{\pm 1.27}$ | $74.64_{\pm 10.86}$ | $97.40_{\pm 2.10}$ | $62.68_{\pm 10.97}$ | $96.79_{\pm 1.94}$ | $37.78_{\pm 15.10}$ | $98.02_{\pm 1.59}$ | $58.81_{\pm 12.17}$ |
| RAIDAR | $85.46_{\pm 0.61}$ | $11.34_{\pm 0.88}$ | $86.69_{\pm 0.64}$ | $8.42_{\pm 0.89}$ | $73.42_{\pm 0.44}$ | $3.06_{\pm 1.50}$ | $64.88_{\pm 0.98}$ | $0.14_{\pm 0.09}$ | $77.61_{\pm 0.45}$ | $5.74_{\pm 0.72}$ |
| RepreGuard | $99.92_{\pm 0.00}$ | $99.42_{\pm 0.11}$ | $99.94_{\pm 0.00}$ | $99.18_{\pm 0.08}$ | $96.78_{\pm 0.10}$ | $77.92_{\pm 1.88}$ | $96.09_{\pm 0.12}$ | $55.84_{\pm 1.11}$ | $98.18_{\pm 0.05}$ | $83.09_{\pm 0.67}$ |
| LTD | $\mathbf{99.99}_{\pm 0.00}$ | $99.66_{\pm 0.39}$ | $99.94_{\pm 0.07}$ | $74.26_{\pm 34.28}$ | $\mathbf{99.91}_{\pm 0.10}$ | $\mathbf{95.36}_{\pm 3.83}$ | $\mathbf{99.97}_{\pm 0.03}$ | $\mathbf{97.32}_{\pm 1.86}$ | $\mathbf{99.95}_{\pm 0.04}$ | $\mathbf{91.65}_{\pm 9.24}$ |
| **LesionProbe** | $99.97_{\pm 0.02}$ | $\mathbf{99.90}_{\pm 0.00}$ | $\mathbf{99.99}_{\pm 0.00}$ | $\mathbf{99.90}_{\pm 0.07}$ | $99.74_{\pm 0.04}$ | $92.40_{\pm 2.17}$ | $97.58_{\pm 0.27}$ | $68.56_{\pm 2.58}$ | $99.32_{\pm 0.07}$ | $90.19_{\pm 0.89}$ |
| *Reference source: Claude-instant* | | | | | | | | | | |
| RoBERTa | $98.26_{\pm 1.73}$ | $54.24_{\pm 24.33}$ | $97.12_{\pm 2.24}$ | $44.50_{\pm 17.88}$ | $95.16_{\pm 1.78}$ | $44.24_{\pm 17.63}$ | $99.56_{\pm 0.23}$ | $86.68_{\pm 12.73}$ | $97.53_{\pm 1.31}$ | $57.42_{\pm 15.18}$ |
| RAIDAR | $85.24_{\pm 0.57}$ | $11.48_{\pm 2.47}$ | $84.66_{\pm 0.69}$ | $1.70_{\pm 0.44}$ | $65.34_{\pm 0.84}$ | $3.42_{\pm 1.08}$ | $72.29_{\pm 0.13}$ | $1.34_{\pm 0.26}$ | $76.88_{\pm 0.48}$ | $4.49_{\pm 0.98}$ |
| RepreGuard | $99.92_{\pm 0.01}$ | $\mathbf{98.42}_{\pm 0.30}$ | $\mathbf{99.89}_{\pm 0.01}$ | $\mathbf{97.70}_{\pm 0.22}$ | $95.15_{\pm 0.11}$ | $66.50_{\pm 1.27}$ | $98.34_{\pm 0.14}$ | $75.72_{\pm 1.25}$ | $98.33_{\pm 0.06}$ | $84.59_{\pm 0.41}$ |
| LTD | $99.83_{\pm 0.11}$ | $98.12_{\pm 1.16}$ | $96.24_{\pm 0.89}$ | $71.22_{\pm 8.21}$ | $87.59_{\pm 1.18}$ | $53.18_{\pm 6.88}$ | $99.89_{\pm 0.02}$ | $97.56_{\pm 4.62}$ | $95.89_{\pm 0.53}$ | $80.02_{\pm 3.07}$ |
| **LesionProbe** | $\mathbf{99.96}_{\pm 0.02}$ | $98.38_{\pm 1.14}$ | $\mathbf{99.89}_{\pm 0.09}$ | $97.00_{\pm 2.12}$ | $\mathbf{97.42}_{\pm 1.46}$ | $\mathbf{76.88}_{\pm 6.61}$ | $\mathbf{99.96}_{\pm 0.05}$ | $\mathbf{99.04}_{\pm 1.28}$ | $\mathbf{99.31}_{\pm 0.38}$ | $\mathbf{92.83}_{\pm 2.10}$ |

**Experimental Protocol**. We use Llama-3.1-8B as the default proxy for LesionProbe.Sampled labeled HWT-LGT pairs serve as reference data for fitting or calibrating LesionProbe, RoBERTa, RAIDAR, RepreGuard, and LTD, following their respective implementations. For cross-LLM evaluation, reference pairs come from one generation source, and evaluation covers all four sources. For each reference source, a run samples 512 pairs from a pool of 1,800 pairs; each test source contains 1,000 held-out pairs. The reference source defines the ID setting, and the other three define OOD settings. For cross-domain evaluation, reference data come from three domains, and OOD evaluation uses the held-out domain; each domain is held out in turn. Cross-benchmark evaluation uses reference data from one benchmark and tests on all five, yielding five ID and 20 OOD settings. The proxy-model, text-length, transformation, and sparse-layer experiments use mixed-source, mixed-domain data. Each run samples 512 reference pairs and evaluates on a disjoint mixed test set of 1,024 pairs from the same sources and domains. Paraphrasing and perturbations are applied only to held-out test samples. Channel, detector-branch, and loss ablations follow the cross-LLM and cross-domain protocols. For these reference-based methods, we repeat reference-data sampling and method-specific fitting or calibration five times and report means and standard deviations. These methods share the sampled reference pairs and test set within each run. Matched HWT–LGT pairs remain together during splitting. Likelihood, LRR, FastDetect-GPT, and Binoculars require no reference data and are evaluated once per test set. Further protocol details are provided in Appendix B.3.

### 4.2 RQ1: Effectiveness and Generalization

We first examine whether module contribution patterns support effective detection and remain informative under distribution shifts. Table 1 and Figure 3 report cross-LLM and leave-one-domain-out evaluations, respectively. Cross-benchmark experiments in Appendix C further assess transfer across different data collections. These evaluations yield two main observations.

(1) LesionProbe improves average cross-source detection while maintaining competitive ID performance. Its ID TPR closely matches LTD (97.76% versus 97.78%), whereas its average OOD TPR

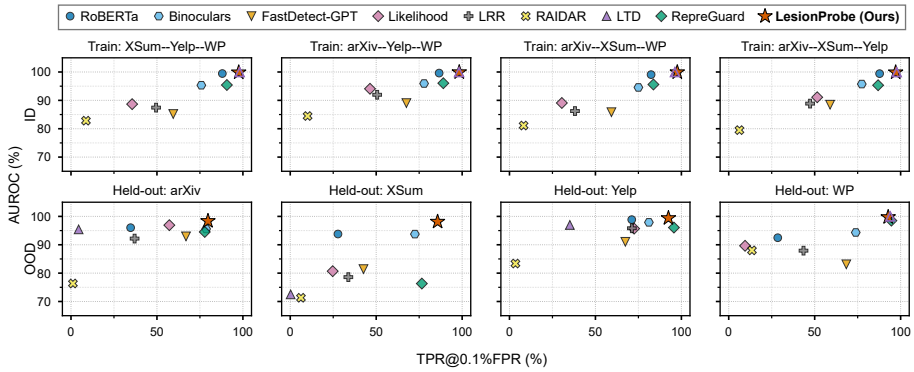


Figure 3: **Leave-one-domain-out generalization.** Columns correspond to held-out domains: arXiv, XSum, Yelp, and Writing Prompts (WP). Top: ID performance averaged over the three reference domains. Bottom: OOD performance on the held-out domain.

reaches $87.80\%$, compared with $83.80\%$ for the strongest competing method, RepreGuard. LesionProbe also achieves the highest average OOD AUROC of $98.74\%$ among the compared methods. The separation between comparable ID results and stronger average OOD results supports the transferability of module contribution patterns across generation sources. This advantage is not uniform across source pairs: Claude-instant remains a challenging transfer target, and LTD performs better on some transfers to this source.

(2) The detection advantage extends to changes in writing domain. LesionProbe achieves the best average performance on held-out domains, with $98.92\%$ AUROC and $87.66\%$ TPR, while retaining ID performance comparable to LTD. Its average OOD TPR exceeds the strongest competing baseline, RepreGuard, by $1.49$ percentage points. The improvements are particularly evident on held-out arXiv and XSum, where LesionProbe leads on both metrics. Together with the cross-source results, these findings show that the patterns remain useful under two separately evaluated forms of distribution shift, although LesionProbe does not lead on every domain and metric.

### 4.3 RQ2: Contribution of Lesion Responses

We next examine which response information and detector components contribute to performance. The ablations in Appendix D vary the lesion-response channels, detector branches, and training objective under the cross-LLM and cross-domain protocols. They reveal two findings.

(1) Combining lesion responses improves detection over individual response channels. Each lesion configuration provides discriminative information, but combining attention, MLP, and joint responses yields the highest mean TPR across all 16 source-training/test combinations. On held-out arXiv, the combined representation reaches $79.68\%$ TPR, compared with $65.80\%$ for the strongest individual channel. The benefit is therefore not limited to ID discrimination: using multiple response types also improves detection under distribution shifts. These comparisons support using multiple lesion responses to capture detection evidence that an individual response channel may miss.

(2) Sequential patterns and statistical summaries provide complementary detection evidence. The sequence branch is more effective than the statistics branch on both metrics across all four held-out domains, whereas the statistics branch is particularly competitive on cross-source transfers from Claude-instant. Combining the branches improves mean TPR over either branch alone in all 12 source-training/test combinations trained on ChatGPT, PaLM, or Llama. The dual-branch detector does not dominate every setting, but it benefits from combining representations whose relative strengths vary across distribution shifts.

### 4.4 RQ3: Sensitivity to Proxy and Input Conditions

We evaluate sensitivity to proxy choice, text length, and text transformations using disjoint reference and test sets with shared source and domain coverage. Each proxy uses a separately trained detector, and transformations affect only test texts.

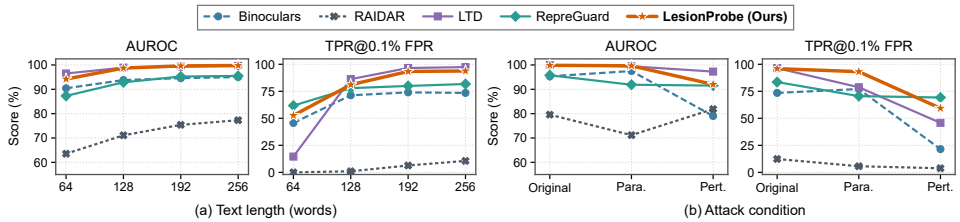


Figure 4: **Robustness to text length and attacks.** **(a)** Detection across text lengths (in words). **(b)** Detection on original (Original), DIPPER-paraphrased (Para.), and perturbed (Pert.) texts from DetectRL (Wu et al., 2024). Each panel reports AUROC (left) and TPR@0.1%FPR (right).

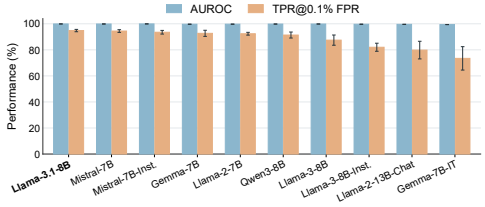


Figure 5: **Performance across proxy LLMs.** AUROC and TPR@0.1%FPR are evaluated on the mixed-source ID test set. Results are reported as mean ± standard deviation over five runs.

Table 2: **Sparse layer interventions.** Results are mean ± std over five runs; percentages in parentheses denote relative Transformer-block computation cost. Peak-5: $\{1, 2, 3, 31, 32\}$; Middle-5: $\{14, 15, 16, 17, 18\}$.

| Layers | Runs | Cost(S) | AUROC | TPR@0.1 |
|---|---|---|---|---|
| All | 97 | 1616 (100%) | $99.88_{\pm 0.04}$ | $94.80_{\pm 0.90}$ |
| Peak-5 | 16 | 320 (19.8%) | $99.35_{\pm 0.14}$ | $84.73_{\pm 2.78}$ |
| Middle-5 | 16 | 287 (17.8%) | $99.04_{\pm 0.31}$ | $83.95_{\pm 2.68}$ |
| $\{1, 32\}$ | 7 | 131 (8.1%) | $99.13_{\pm 0.33}$ | $74.61_{\pm 7.66}$ |
| $\{32\}$ | 4 | 35 (2.2%) | $98.26_{\pm 0.70}$ | $57.99_{\pm 1.41}$ |

**Proxy Choice**. Figure 5 shows effective detection across all ten evaluated proxies under the mixed-source ID protocol. Mean AUROC ranges from $99.40\%$ to $99.91\%$, supporting the applicability of our approach across different proxy families.

**Text Length and Transformations**. Figure 4 reveals that LesionProbe remains competitive on short texts, while longer inputs improve low-FPR detection. It ranks second in TPR at all tested length and achieves the highest AUROC at 192 and 256 words. Meanwhile, LesionProbe is more robust to paraphrasing than to perturbations. It achieves the highest AUROC ($99.66\%$) and TPR ($93.15\%$) among the compared methods on paraphrased texts.

### 4.5 RQ4: Computation-Performance Trade-off

We finally examine how reducing the number of intervened layers affects extraction cost and detection. Peak-5 selects layers with larger training-set Shapley differences, while Middle-5 provides a five-layer comparison. For a selected layer set $S$, extraction uses $1 + 3|S|$ runs. Since cached upstream states can be reused, we measure cost by the total number of evaluated Transformer blocks, as defined in Appendix B.4. This measure characterizes block computation rather than wall-clock runtime. As shown in Table 2, Sparse interventions retain strong overall discrimination at substantially reduced block-computation cost. Peak-5 uses $19.8\%$ of the full-layer cost, with an AUROC decrease of only $0.53$ percentage points. Peak-5 slightly outperforms Middle-5 on both metrics, although the margin between the two five-layer subsets is small. Both configurations retain high AUROC, showing that substantial overall discrimination can be preserved with sparse extraction.

## 5 Conclusion

We present LesionProbe, a framework for LGT detection through a frozen proxy LLM's responses to component interventions. Individual and joint ablations of attention and MLP branches reveal module contribution patterns that distinguish HWT from LGT, while Shapley attribution characterizes their layer-wise differences. Experiments demonstrate competitive ID detection and improved average generalization to unseen generation sources and writing domains. These findings support module contribution patterns as transferable detection signals, while sparse-layer evaluations reveal a trade-off between extraction cost and low-FPR detection sensitivity.

## AI USE STATEMENT

The authors used generative AI tools for light language polishing and literature retrieval. AI assistance was limited to improving grammar, clarity, and academic expression, and supporting the discovery of relevant literature. Generative AI was not used for the substantive development or evaluation of this work. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the manuscript.

## ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. We study AI-generated text detection using existing public benchmarks, without recruiting human participants or collecting new personal data. Our research aims to support transparency and responsible use of language models. We emphasize responsible use of detection tools, with respect for user privacy, fair treatment of authors, and the context in which texts are evaluated.

## REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the complete implementation of LESIONPROBE as part of the supplementary materials at submission. The submitted code includes module lesion and contribution extraction from frozen proxy LLMs, normalized Jensen–Shannon divergence computation, feature preprocessing, detector training and evaluation, and Shapley-based attribution analysis. We also provide the configurations and scripts for the main experiments, including cross-LLM and cross-domain evaluation, robustness analyses, proxy-model comparisons, and ablation studies.

Experimental details, including data splits, random seeds, optimization hyperparameters, and evaluation settings, are documented to support reproduction of the reported results. The code package submitted for review is anonymized in accordance with the double-blind review policy. Upon publication, we will publicly release the complete codebase and reproducibility materials in a public repository.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Releasing Claude Instant 1.2. `https://www.anthropic.com/news/releasing-claude-instant-1-2`, 2023. Accessed: 2026-08-14.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *International Conference on Learning Representations*, volume 2024, pp. 24814–24836, 2024.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S Chao, and Derek F Wong. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *Transactions of the Association for Computational Linguistics*, 13:1812–1831, 2025.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12463–12492, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-Jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations*, pp. 111–116, 2019.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 5484–5495, 2021.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, 2023.

Google. Introducing PaLM 2. `https://blog.google/innovation-and-ai/products/google-palm-2-ai-large-language-model/`, 2023. Accessed: 2026-08-14.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*, 2024.

Harold W Kuhn and Albert William Tucker. *Contributions to the Theory of Games*. Number 28. Princeton University Press, 1953.

Yafu Li, Qintong Li, Leyang Cui, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, Yue Zhang, et al. 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)*, pp. 36–53, 2024.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative ai detection via rewriting. In *International Conference on Learning Representations*, volume 2024, pp. 6013–6030, 2024.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pp. 24950–24962. PMLR, 2023.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 1797–1807, 2018.

OpenAI. Introducing ChatGPT. `https://openai.com/index/chatgpt/`, 2022. Accessed: 2026-08-14.

Dezhi Ran, Yuan Cao, Mengzhou Wu, Simin Chen, Yuzhe Guo, Jun Ren, Zihe Song, Hao Yu, Jialei Wei, Linyi Li, et al. From assistant to independent developer—are gpts ready for software development? In *International Conference on Learning Representations*, volume 2026, pp. 100719–100739, 2026.

Chris Rorden and Hans-Otto Karnath. Using human brain lesions to infer function: a relic from a past era in the fmri age? *Nature Reviews Neuroscience*, 5(10):812–819, 2004.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052, 2023.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12395–12412, 2023.

Zhen Sun, Zongmin Zhang, Xinyue Shen, Ziyi Zhang, Yule Liu, Michael Backes, Yang Zhang, and Xinlei He. Are we in the ai-generated text world already? quantifying and monitoring aigt on social media. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22975–23005, 2025a.

Zhongxiang Sun, Qipeng Wang, Haoyu Wang, Xiao Zhang, and Jun Xu. Detection and mitigation of hallucination in large reasoning models: A mechanistic perspective. *arXiv preprint arXiv:2505.12886*, 2025b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alistair Turcan, Kexin Huang, Lei Li, and Martin J Zhang. Tusoai: agentic optimization for scientific methods. In *International Conference on Learning Representations*, volume 2026, pp. 50077–50125, 2026.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1702–1717, 2024.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3964–3992, 2024.

Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S Chao. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401, 2024.

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

Dizhan Xue, Jing Cui, Shengsheng Qian, Chuanrui Hu, and Changsheng Xu. Some: A realistic benchmark for llm-based social media agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1391–1399, 2026.

Jingheng Ye, Shen Wang, Deqing Zou, Yibo Yan, Kun Wang, Hai-Tao Zheng, Ruitong Liu, Zenglin Xu, Irwin King, Philip S Yu, et al. Position: Llms can be good tutors in english education. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17527–17546, 2025.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Hongyi Zhou, Jin Zhu, Kai Ye, Ying Yang, Erhan Xu, and Chengchun Shi. Learn-to-distance: Distance learning for detecting llm-generated text. *arXiv preprint arXiv:2601.21895*, 2026.

The appendices provide supplementary methodological details and experimental analyses. Appendix A describes the input formulation. Appendix B provides dataset details, baseline implementations, evaluation protocols, and computation accounting for sparse interventions. Appendix C evaluates cross-benchmark generalization. Appendix D presents ablations on lesion-response channels, detector branches, and classification losses. Appendix E examines sensitivity to test-set size and quantifies test-sampling uncertainty under ID and OOD evaluation.

# A Input Formulation

The proxy LLM directly processes the tokenized text sequence without any additional instruction or task-specific prompt. Given a text $x = (x_1, \ldots, x_n)$, the autoregressive proxy model produces a next-token predictive distribution at each position,

$$p_\theta(\cdot \mid x_{\le t}), \qquad t = 1, \ldots, n. \tag{11}$$

Thus, each text induces a sequence of context-conditioned predictions determined by its preceding tokens.

LesionProbe examines how these predictions change when internal components of the same frozen proxy model are selectively removed. For each text, we compare the intact predictive distributions with those obtained under attention, MLP, and joint lesions at individual layers. The resulting intervention responses therefore characterize how the proxy model's predictions depend on its internal components for that particular input. Our detector learns whether the resulting layer- and token-wise response patterns systematically differ between HWT and LGT.

# B Experimental Details

This section provides additional details for the main experiments. We describe the DetectRL dataset, baseline implementations, and evaluation protocols used across different experimental settings. We also define the computational-cost measure used in the sparse-layer efficiency analysis.

## B.1 Experimental Setup

**Dataset Details.** We conduct the main evaluations on the DetectRL benchmark (Wu et al., 2024), which contains human-written text (HWT) and LLM-generated text (LGT) across four writing domains: arXiv abstracts, XSum, Writing Prompts, and Yelp reviews. The LGT samples are generated by four source LLMs: GPT-3.5-turbo, Claude-instant, PaLM-2-bison, and Llama-2-70B. For each domain, each source contributes 700 LGT samples, together with 2,800 corresponding HWT samples. We use three conditions from DetectRL: Direct for the main experiments, DIPPER Paraphrase (DP) for paraphrasing robustness, and Sentence-Level Perturbation (SP) for perturbation robustness. Table 3 summarizes the data used in our experiments.

## B.2 Baseline Implementation Details

We follow the original implementations and adopt the recommended or best-performing configurations reported in the corresponding works where applicable.

**Textual Features.** For RoBERTa, we fully fine-tune RoBERTa-base as a binary classifier.

**Predictive Statistics.** These methods do not use the sampled labeled reference pairs. For Likelihood, we use GPT-2 and compute the mean token log-likelihood as the detection score. LRR uses GPT-Neo-2.7B to compute the log-likelihood–log-rank ratio. For FastDetect-GPT, we use GPT-J-6B as the sampling model and GPT-Neo-2.7B as the scoring model, following the reported black-box configuration. Binoculars uses Falcon-7B as the observer and Falcon-7B-Instruct as the performer, both evaluated in `bfloat16`.

**Responses to Input Transformations.** RAIDAR primarily uses GPT-3.5-Turbo for rewriting, while the original study also reports competitive detection performance using Llama-2-7B-Chat. As GPT-3.5-Turbo was unavailable in our experimental environment, we use a locally deployed Llama-2-7B-Chat model while retaining the original invariance features and logistic regression classifier.

Table 3: **Statistics of the DetectRL data used in our experiments.** GPT, Claude, PaLM, and Llama denote GPT-3.5-turbo, Claude-instant, PaLM-2-bison, and Llama-2-70B, respectively.

| Condition | Domain | HWT | GPT | Claude | PaLM | Llama | Total |
|---|---|---|---|---|---|---|---|
| Direct | arXiv Abstracts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | XSum | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Writing Prompts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Yelp Reviews | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | **Subtotal** | **11,200** | **2,800** | **2,800** | **2,800** | **2,800** | **22,400** |
| DP | arXiv Abstracts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | XSum | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Writing Prompts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Yelp Reviews | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | **Subtotal** | **11,200** | **2,800** | **2,800** | **2,800** | **2,800** | **22,400** |
| SP | arXiv Abstracts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | XSum | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Writing Prompts | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | Yelp Reviews | 2,800 | 700 | 700 | 700 | 700 | 5,600 |
| | **Subtotal** | **11,200** | **2,800** | **2,800** | **2,800** | **2,800** | **22,400** |
| **Total** | | **33,600** | **8,400** | **8,400** | **8,400** | **8,400** | **67,200** |

For LTD, we use Gemma-2-9B-IT with LoRA adaptation and the learned reconstruction-distance detector following the official implementation.

**Internal Representations.** RepreGuard uses Llama-3.1-8B as the representation model and trains its probe following the original implementation.

### B.3 Experimental Protocol Details

**Repeated runs and reporting.** For LesionProbe and the reference-based baselines, we conduct five runs per experimental setting. Each run independently samples labeled reference pairs and repeats the method-specific fitting or calibration procedure. Within each run, these methods share the sampled reference pairs and held-out test set. Matched HWT–LGT pairs remain together during splitting, and reference and test sets are strictly disjoint. We report means and standard deviations across the five runs. Reference-free methods are evaluated once per test set because their scores do not depend on reference-data sampling.

**Cross-LLM evaluation.** For each source LLM, we construct a reference pool of 1,800 matched HWT–LGT pairs and a disjoint test set of 1,000 pairs. Each run independently samples 512 reference pairs from the pool for method-specific fitting or calibration. Evaluation covers all four test sources, with the reference source defining the ID setting and the other three defining OOD settings. Test pairs are never used for fitting or calibration.

**Cross-domain evaluation.** We follow a leave-one-domain-out protocol over the four writing domains. For each split, reference data come from three domains, and the remaining domain is held out for OOD evaluation. Each domain serves as the held-out domain in turn. Reference-based methods follow the five-run procedure described above, sharing reference samples and test sets within each run.

**Mixed-source evaluation.** The proxy-model, text-length, transformation, and sparse-layer experiments use mixed-source, mixed-domain data. Each run samples 512 matched reference pairs from a pool spanning all four generation sources and writing domains. Evaluation uses a disjoint mixed test set of 1,024 matched pairs covering the same sources and domains. Reference-based methods follow the same five-run procedure, with results reported as mean $\pm$ standard deviation. Paraphras-

ing and perturbations are applied only to held-out test samples; transformed samples are never used for fitting or calibration.

### B.4 Computation Accounting for Sparse Interventions

Sparse interventions affect Transformer computation through both the number and positions of selected layers. Selecting fewer layers reduces the number of intervention runs, while selecting later layers shortens the recomputed model suffix. By contrast, for fixed input length and output positions, LM-head projections and distribution comparisons, including JSD, depend primarily on the number of runs rather than intervention depth. We therefore use Transformer-block counts to capture both effects under the same proxy model and input length.

Let $S \subseteq \{1, \ldots, L\}$ denote the selected intervention layers. One intact forward pass provides shared reference predictions and cached hidden states. Each selected layer receives three interventions: Attention, MLP, and joint lesions. An intervention at layer $s$ reuses the intact states preceding that layer and recomputes layers $s$ through $L$ over the entire sequence. Each run produces predictions for all valid token positions. The total number of runs is

$$\mathrm{Runs}(S) = 1 + 3|S|, \tag{12}$$

including one intact pass and three intervention runs per selected layer.

We count one execution of a Transformer block over the input sequence as one computation unit. The total block count is

$$\mathrm{Cost}(S) = L + 3\sum_{s \in S}(L - s + 1), \tag{13}$$

where the first term accounts for the intact pass and the second for the intervention passes. Relative cost is normalized by full-layer extraction:

$$\text{Rel. Cost}(S) = \frac{\mathrm{Cost}(S)}{\mathrm{Cost}(\{1, \ldots, L\})} \times 100\%. \tag{14}$$

For the 32-layer proxy, full-layer extraction requires 97 runs and 1,616 block executions. Peak-5 and Middle-5 each require 16 runs but incur 320 and 287 block executions, respectively, illustrating the effect of intervention depth at a fixed run count.

## C Cross-Benchmark Generalization

Table 4: **Composition of the selected benchmark subsets.** All selected texts are in English. Domains and source LLMs refer to the subsets used in our experiments, rather than the full benchmarks. CMV = ChangeMyView; ELI5 = Explain Like I'm Five; WP = WritingPrompts; NYT = The New York Times; IMDb = Internet Movie Database; XSum = Extreme Summarization. LLaMA-13B and LLaMA-65B are first-generation base models. AIGTBench uses AWQ-quantized versions of LLaMA-65B and Llama-3.1-70B-Instruct. Cohere retains its dataset label because the consulted sources do not specify an exact API version or checkpoint.

| Benchmark | Domains | Source LLMs |
|---|---|---|
| RAID (Dugan et al., 2024) | arXiv, NYT, Reddit, IMDb | gpt-3.5-turbo-0613, gpt-4-0613, Mistral-7B-v0.1, MPT-30B |
| MAGE (Li et al., 2024) | CMV, ELI5, XSum, Yelp | LLaMA-13B, GLM-130B, BLOOM-7B1, FLAN-T5-XXL |
| M4GT-Bench (Wang et al., 2024) | arXiv, Reddit (ELI5), WikiHow, Wikipedia | BLOOMZ-176B, Cohere, text-davinci-003, Dolly-v2-12B |
| AIGTBench (Sun et al., 2025a) | Medium, Quora, Reddit | GPT-4o-mini, Llama-2-70B-Chat, Llama-3.1-70B-Instruct, LLaMA-65B |
| DetectRL (Wu et al., 2024) | arXiv, XSum, Yelp, WP | GPT-3.5-Turbo, Llama-2-70b-chat-hf, PaLM-2 (chat-bison@002), Claude-instant-1.2 |

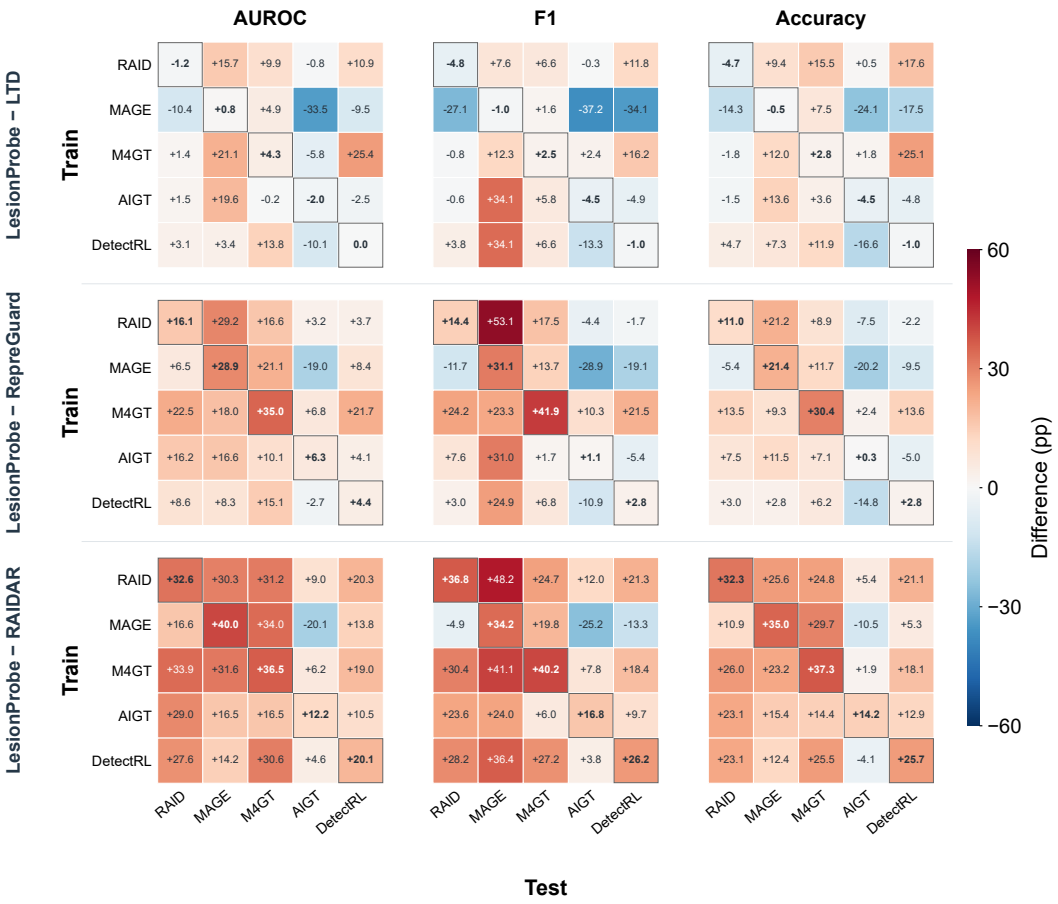


Figure 6: **Cross-benchmark performance differences between LesionProbe and strong baselines.** Each cell shows the performance difference (LesionProbe − baseline, in percentage points) in AUROC, F1, or Accuracy. Rows indicate the training benchmark and columns indicate the test benchmark. Outlined diagonal cells correspond to ID evaluation, while off-diagonal cells represent OOD cross-benchmark transfer. Positive values indicate higher performance for LesionProbe. Results are averaged over five independent runs.

**Experimental Setup.** To complement the main experiments conducted on DetectRL, we further evaluate cross-benchmark generalization across five benchmarks: RAID, MAGE, M4GT-Bench, AIGTBench, and DetectRL. As summarized in Table 4, the selected English subsets cover 19 benchmark-specific domain subsets and 20 source-LLM subsets, providing substantially broader variation in writing domains, generation sources, and dataset composition. To maintain a consistent binary detection task, we retain only original HWT and LGT, excluding rewritten, adversarially perturbed, and mixed human–AI texts. For each benchmark, we select up to four domains and four source LLMs and construct non-overlapping training and test sets of 4,000 and 2,000 texts, respectively. Both splits are class-balanced, with approximately balanced domain coverage and generator representation within LGT. Where HWT–LGT pairings are available, both members are assigned to the same split; otherwise, the two classes are sampled independently.

**Evaluation Protocol.** To provide a broader assessment of conventional classification performance, we compare LesionProbe with selected strong baselines from the main experiments using Accuracy, F1, and AUROC. The compared baselines are LTD, RepreGuard, and RAIDAR. LGT is treated as the positive class when computing F1. Each detector is trained on one benchmark and evaluated on all five test sets, yielding five ID and 20 OOD settings. For Accuracy and F1, LesionProbe and RAIDAR use a probability threshold of 0.5, whereas RepreGuard uses its transferred threshold. For LTD, we select a threshold for each training benchmark and seed by maximizing F1 on training scores. All thresholds remain fixed across target benchmarks without tuning on target test labels.

LesionProbe uses a frozen Llama-3.1-8B proxy, with its detector architecture and optimization settings held fixed across benchmarks. All metrics are computed separately for each run and averaged over five independent runs.

Figure 6 compares LesionProbe with the selected baselines across all training–test benchmark combinations. Each cell reports the performance difference between LesionProbe and the corresponding baseline in percentage points; diagonal and off-diagonal cells denote ID and OOD evaluation, respectively.

**ID Performance.** Across the five ID settings, LesionProbe achieves an average AUROC of 97.52%, compared with 97.14% for LTD, while RepreGuard and RAIDAR obtain 79.37% and 69.23%, respectively. LesionProbe performs particularly well on MAGE and M4GT-Bench, and closely matches LTD on DetectRL (99.88% versus 99.90%). For F1 and Accuracy, LTD achieves higher averages (94.23% and 94.14%) than LesionProbe (92.47% and 92.55%). Thus, the two methods show comparable overall ID performance, with LTD retaining an advantage in the threshold-dependent metrics.

**OOD Performance.** Averaged over the 20 off-diagonal transfer settings, LesionProbe achieves AUROC, F1, and Accuracy scores of 84.08%, 74.00%, and 76.03%, respectively, the highest averages among the compared methods. Relative to LTD, these correspond to gains of 2.90, 1.23, and 2.49 percentage points, with larger average gains over RepreGuard and RAIDAR. LesionProbe also exhibits a smaller macro-average ID–OOD gap than LTD across all three metrics, indicating better preservation of performance under benchmark shifts.

The gains, however, are not uniform across transfer directions. AIGTBench is a particularly challenging target, most notably for MAGE→AIGTBench, where LesionProbe underperforms LTD by 33.5, 37.2, and 24.1 percentage points in AUROC, F1, and Accuracy, respectively. These results indicate stronger average cross-benchmark transfer for LesionProbe, while also revealing substantial source–target asymmetry.

# D Ablation Studies

This section provides additional analyses of the main components involved in LesionProbe detection. We first examine the contribution of the three lesion-response channels and the roles of the sequence and statistics branches in the detector. We then compare focal loss with standard binary cross-entropy to assess the effect of the classification objective.

## D.1 Ablation on Lesion-Response Channels

**Experimental Setup.** We assess the contribution and complementarity of $D_A$, $D_M$, and $D_0$, corresponding to attention, MLP, and joint lesions, respectively. We compare five input configurations: each channel individually, the two-channel representation $(D_A, D_M)$, and the full representation $(D_A, D_M, D_0)$. The two-channel configuration provides a reference for evaluating the additional predictive value of the joint-lesion response beyond the individual module responses. All variants use Llama-3.1-8B as the proxy model and share the same detector design, training objective, data splits, and optimization settings, with only the input channels varied. We follow the same cross-LLM and leave-one-domain-out cross-domain protocols as the main experiments and report mean $\pm$ standard deviation over five independent runs.

**Cross-LLM Results.** Table 5 shows that each individual lesion channel contains discriminative information, while combining all three generally improves performance across generation sources. The full representation achieves higher mean TPR than every single-channel variant in all 16 training–test settings. For example, when trained on Claude and evaluated on Llama-2-70B, it achieves 99.89% AUROC and 97.00% TPR. The corresponding AUROC/TPR values are 99.50%/78.86% for $D_A$, 98.44%/42.16% for $D_M$, and 99.31%/59.62% for $D_0$. Relative to $(D_A, D_M)$, the full representation also achieves higher average OOD AUROC (98.74% versus 98.52%) and TPR (87.80% versus 83.04%), suggesting additional predictive value from $D_0$. The advantage is not uniform across metrics: for Claude-to-PaLM transfer, $(D_A, D_M)$ achieves higher AUROC (98.42% versus 97.42%), but lower TPR (67.18% versus 76.88%). Overall, these com-

Table 5: **Contribution-channel ablation across LLM sources.** Each cell reports AUROC / TPR@0.1%FPR (mean ± std, %) over five runs. Within each training-source block, the best AUROC and TPR for each test source are independently highlighted in blue and bolded; gray rows denote the full three-channel representation.

| Train | Channels | ChatGPT | Claude-instant | Google-PaLM | Llama-2-70B |
|---|---|---|---|---|---|
| ChatGPT | Attention ($D_A$) | **99.96**$_{\pm 0.03}$ / 99.62$_{\pm 0.08}$ | 94.20$_{\pm 0.17}$ / 52.76$_{\pm 7.29}$ | 98.07$_{\pm 0.28}$ / 85.26$_{\pm 0.54}$ | 99.95$_{\pm 0.03}$ / 99.02$_{\pm 0.23}$ |
| | MLP ($D_M$) | 99.89$_{\pm 0.01}$ / 99.48$_{\pm 0.28}$ | 91.20$_{\pm 0.87}$ / 44.58$_{\pm 5.61}$ | 97.73$_{\pm 0.49}$ / 78.14$_{\pm 3.67}$ | 99.91$_{\pm 0.03}$ / 97.52$_{\pm 0.40}$ |
| | Joint ($D_0$) | 99.92$_{\pm 0.03}$ / 99.60$_{\pm 0.16}$ | 94.34$_{\pm 0.82}$ / 49.74$_{\pm 8.13}$ | 98.19$_{\pm 0.30}$ / 81.94$_{\pm 3.02}$ | 99.97$_{\pm 0.01}$ / 98.12$_{\pm 0.51}$ |
| | Attention + MLP ($D_A, D_M$) | 99.95$_{\pm 0.03}$ / **99.88**$_{\pm 0.04}$ | 94.84$_{\pm 0.68}$ / 55.84$_{\pm 5.71}$ | **99.03**$_{\pm 0.09}$ / 85.06$_{\pm 4.00}$ | **99.99**$_{\pm 0.01}$ / **99.62**$_{\pm 0.08}$ |
| | **Full** ($D_A, D_M, D_0$) | 99.95$_{\pm 0.04}$ / 99.86$_{\pm 0.05}$ | **95.66**$_{\pm 0.70}$ / **67.36**$_{\pm 2.56}$ | 98.81$_{\pm 0.26}$ / **87.20**$_{\pm 1.52}$ | **99.99**$_{\pm 0.01}$ / 99.60$_{\pm 0.29}$ |
| Claude | Attention ($D_A$) | 99.53$_{\pm 0.55}$ / 82.56$_{\pm 22.29}$ | 99.48$_{\pm 0.42}$ / 82.42$_{\pm 14.52}$ | 97.86$_{\pm 1.14}$ / 63.18$_{\pm 17.68}$ | 99.50$_{\pm 0.58}$ / 78.86$_{\pm 22.13}$ |
| | MLP ($D_M$) | 98.58$_{\pm 0.41}$ / 23.18$_{\pm 9.33}$ | 99.73$_{\pm 0.08}$ / 94.56$_{\pm 5.52}$ | 93.61$_{\pm 1.79}$ / 25.90$_{\pm 9.16}$ | 98.44$_{\pm 0.62}$ / 42.16$_{\pm 7.72}$ |
| | Joint ($D_0$) | 99.48$_{\pm 0.31}$ / 69.20$_{\pm 15.19}$ | 99.90$_{\pm 0.05}$ / 98.80$_{\pm 1.04}$ | 95.54$_{\pm 1.03}$ / 42.82$_{\pm 9.52}$ | 99.31$_{\pm 0.39}$ / 59.62$_{\pm 17.78}$ |
| | Attention + MLP ($D_A, D_M$) | 99.84$_{\pm 0.09}$ / 93.48$_{\pm 9.62}$ | 99.76$_{\pm 0.18}$ / 92.98$_{\pm 6.52}$ | **98.42**$_{\pm 1.18}$ / 67.18$_{\pm 22.31}$ | 99.83$_{\pm 0.22}$ / 90.00$_{\pm 12.55}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.96**$_{\pm 0.02}$ / **98.38**$_{\pm 1.14}$ | **99.96**$_{\pm 0.05}$ / **99.04**$_{\pm 1.28}$ | 97.42$_{\pm 1.46}$ / **76.88**$_{\pm 6.61}$ | **99.89**$_{\pm 0.09}$ / **97.00**$_{\pm 2.12}$ |
| PaLM | Attention ($D_A$) | **99.98**$_{\pm 0.02}$ / 99.10$_{\pm 0.70}$ | 94.64$_{\pm 1.19}$ / 46.24$_{\pm 10.56}$ | 99.49$_{\pm 0.04}$ / 89.86$_{\pm 1.67}$ | 99.97$_{\pm 0.02}$ / 98.62$_{\pm 0.81}$ |
| | MLP ($D_M$) | 99.89$_{\pm 0.01}$ / 98.78$_{\pm 1.41}$ | 92.38$_{\pm 0.69}$ / 40.16$_{\pm 5.78}$ | 99.03$_{\pm 0.05}$ / 81.60$_{\pm 2.90}$ | 99.90$_{\pm 0.03}$ / 97.00$_{\pm 1.52}$ |
| | Joint ($D_0$) | 99.97$_{\pm 0.02}$ / 99.54$_{\pm 0.21}$ | 95.92$_{\pm 0.65}$ / 55.60$_{\pm 5.48}$ | 99.37$_{\pm 0.06}$ / 86.54$_{\pm 3.25}$ | **99.99**$_{\pm 0.01}$ / 99.08$_{\pm 0.40}$ |
| | Attention + MLP ($D_A, D_M$) | 99.93$_{\pm 0.02}$ / **99.90**$_{\pm 0.00}$ | 96.21$_{\pm 0.65}$ / 56.62$_{\pm 5.71}$ | 99.71$_{\pm 0.06}$ / 89.04$_{\pm 2.77}$ | **99.99**$_{\pm 0.00}$ / 99.58$_{\pm 0.16}$ |
| | **Full** ($D_A, D_M, D_0$) | 99.97$_{\pm 0.02}$ / **99.90**$_{\pm 0.00}$ | **97.58**$_{\pm 0.27}$ / **68.56**$_{\pm 2.58}$ | **99.74**$_{\pm 0.04}$ / **92.40**$_{\pm 2.17}$ | **99.99**$_{\pm 0.00}$ / **99.90**$_{\pm 0.07}$ |
| Llama | Attention ($D_A$) | 99.93$_{\pm 0.02}$ / 99.66$_{\pm 0.11}$ | 94.25$_{\pm 0.43}$ / 52.48$_{\pm 4.80}$ | 97.88$_{\pm 0.37}$ / 87.40$_{\pm 0.89}$ | 99.97$_{\pm 0.02}$ / 98.88$_{\pm 0.92}$ |
| | MLP ($D_M$) | 99.89$_{\pm 0.01}$ / 99.68$_{\pm 0.04}$ | 92.51$_{\pm 1.21}$ / 42.62$_{\pm 2.04}$ | 98.15$_{\pm 0.12}$ / 80.24$_{\pm 3.40}$ | 99.96$_{\pm 0.02}$ / 98.00$_{\pm 0.41}$ |
| | Joint ($D_0$) | 99.95$_{\pm 0.04}$ / 99.62$_{\pm 0.13}$ | 95.12$_{\pm 0.41}$ / 54.74$_{\pm 5.58}$ | 98.47$_{\pm 0.19}$ / 84.12$_{\pm 1.99}$ | 99.99$_{\pm 0.00}$ / 98.76$_{\pm 0.15}$ |
| | Attention + MLP ($D_A, D_M$) | 99.94$_{\pm 0.03}$ / **99.88**$_{\pm 0.04}$ | 95.32$_{\pm 0.34}$ / 62.36$_{\pm 3.69}$ | 98.86$_{\pm 0.23}$ / 86.98$_{\pm 3.67}$ | **99.99**$_{\pm 0.00}$ / **99.78**$_{\pm 0.08}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.96**$_{\pm 0.03}$ / 99.86$_{\pm 0.09}$ | **96.64**$_{\pm 0.79}$ / **69.76**$_{\pm 4.54}$ | **98.98**$_{\pm 0.26}$ / **89.24**$_{\pm 0.93}$ | 99.99$_{\pm 0.00}$ / 99.74$_{\pm 0.05}$ |

Table 6: **Contribution-channel ablation across writing domains.** Each cell reports AUROC / TPR@0.1%FPR (mean ± std, %) over five runs. For each leave-one-domain-out setting, the detector is trained on the remaining three domains and evaluated on all four domains. The best AUROC and TPR among the five channel configurations are independently highlighted in blue and bolded; gray rows denote the full three-channel representation.

| Held-out | Channels | Yelp | Writing Prompt | XSum | arXiv |
|---|---|---|---|---|---|
| Yelp | Attention ($D_A$) | 97.08$_{\pm 1.01}$ / 79.90$_{\pm 8.60}$ | 99.79$_{\pm 0.09}$ / 93.02$_{\pm 1.40}$ | 99.62$_{\pm 0.07}$ / 93.00$_{\pm 0.89}$ | 99.89$_{\pm 0.04}$ / 92.94$_{\pm 2.82}$ |
| | MLP ($D_M$) | 98.59$_{\pm 0.23}$ / 84.70$_{\pm 2.44}$ | 99.76$_{\pm 0.08}$ / 93.26$_{\pm 0.59}$ | 99.04$_{\pm 0.16}$ / 89.14$_{\pm 2.26}$ | 98.77$_{\pm 0.18}$ / 66.82$_{\pm 6.62}$ |
| | Joint ($D_0$) | 99.11$_{\pm 0.39}$ / 87.96$_{\pm 3.43}$ | 99.92$_{\pm 0.04}$ / 96.12$_{\pm 1.74}$ | 99.14$_{\pm 0.23}$ / 91.28$_{\pm 1.69}$ | 99.79$_{\pm 0.04}$ / 89.02$_{\pm 3.26}$ |
| | Attention + MLP ($D_A, D_M$) | 98.77$_{\pm 0.27}$ / 87.88$_{\pm 2.14}$ | 99.93$_{\pm 0.03}$ / 97.58$_{\pm 0.39}$ | 99.66$_{\pm 0.11}$ / 94.24$_{\pm 2.85}$ | 99.91$_{\pm 0.06}$ / 95.66$_{\pm 2.59}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.40**$_{\pm 0.17}$ / **92.46**$_{\pm 2.23}$ | **99.96**$_{\pm 0.02}$ / **98.62**$_{\pm 0.73}$ | **99.79**$_{\pm 0.07}$ / **96.00**$_{\pm 0.57}$ | **99.97**$_{\pm 0.03}$ / **98.20**$_{\pm 0.64}$ |
| Writing Prompt | Attention ($D_A$) | 99.65$_{\pm 0.16}$ / 93.00$_{\pm 2.70}$ | 98.90$_{\pm 0.43}$ / 86.24$_{\pm 3.84}$ | 99.68$_{\pm 0.12}$ / 93.90$_{\pm 1.53}$ | 99.89$_{\pm 0.04}$ / 89.70$_{\pm 3.94}$ |
| | MLP ($D_M$) | 99.47$_{\pm 0.10}$ / 90.80$_{\pm 1.35}$ | 98.43$_{\pm 0.56}$ / 78.50$_{\pm 6.71}$ | 99.03$_{\pm 0.13}$ / 87.88$_{\pm 1.24}$ | 98.78$_{\pm 0.20}$ / 72.74$_{\pm 5.31}$ |
| | Joint ($D_0$) | 99.72$_{\pm 0.11}$ / 93.46$_{\pm 1.76}$ | 99.48$_{\pm 0.14}$ / 89.26$_{\pm 3.26}$ | 99.24$_{\pm 0.26}$ / 90.52$_{\pm 1.96}$ | 99.73$_{\pm 0.04}$ / 89.00$_{\pm 1.10}$ |
| | Attention + MLP ($D_A, D_M$) | 99.76$_{\pm 0.06}$ / 95.58$_{\pm 0.79}$ | 99.44$_{\pm 0.06}$ / 88.64$_{\pm 2.38}$ | 99.64$_{\pm 0.08}$ / 94.64$_{\pm 0.90}$ | 99.95$_{\pm 0.01}$ / 96.46$_{\pm 1.74}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.82**$_{\pm 0.02}$ / **97.70**$_{\pm 0.70}$ | **99.75**$_{\pm 0.11}$ / **92.76**$_{\pm 1.23}$ | **99.83**$_{\pm 0.03}$ / **96.40**$_{\pm 0.23}$ | **99.97**$_{\pm 0.01}$ / **97.26**$_{\pm 1.27}$ |
| XSum | Attention ($D_A$) | 99.65$_{\pm 0.14}$ / 91.64$_{\pm 2.65}$ | 99.89$_{\pm 0.05}$ / 95.40$_{\pm 2.62}$ | 96.92$_{\pm 1.17}$ / 74.68$_{\pm 3.56}$ | 99.90$_{\pm 0.04}$ / 93.22$_{\pm 5.42}$ |
| | MLP ($D_M$) | 99.32$_{\pm 0.11}$ / 90.32$_{\pm 1.46}$ | 99.70$_{\pm 0.12}$ / 91.72$_{\pm 0.73}$ | 97.16$_{\pm 0.54}$ / 81.76$_{\pm 2.73}$ | 98.82$_{\pm 0.26}$ / 75.30$_{\pm 6.94}$ |
| | Joint ($D_0$) | 99.57$_{\pm 0.16}$ / 92.24$_{\pm 2.56}$ | 99.90$_{\pm 0.08}$ / 96.64$_{\pm 1.47}$ | 97.00$_{\pm 1.14}$ / 80.98$_{\pm 5.63}$ | 99.81$_{\pm 0.03}$ / 88.46$_{\pm 3.89}$ |
| | Attention + MLP ($D_A, D_M$) | 99.72$_{\pm 0.08}$ / 93.96$_{\pm 0.64}$ | 99.93$_{\pm 0.02}$ / 97.50$_{\pm 0.95}$ | 96.01$_{\pm 1.31}$ / 78.26$_{\pm 3.74}$ | 99.95$_{\pm 0.04}$ / 97.38$_{\pm 1.29}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.81**$_{\pm 0.03}$ / **96.70**$_{\pm 1.22}$ | **99.96**$_{\pm 0.03}$ / **98.82**$_{\pm 0.54}$ | **98.13**$_{\pm 0.60}$ / **85.72**$_{\pm 4.01}$ | **99.99**$_{\pm 0.01}$ / **99.40**$_{\pm 0.66}$ |
| arXiv | Attention ($D_A$) | 99.59$_{\pm 0.07}$ / 92.94$_{\pm 0.89}$ | 99.82$_{\pm 0.05}$ / 94.14$_{\pm 0.94}$ | 99.56$_{\pm 0.03}$ / 90.82$_{\pm 1.99}$ | 96.67$_{\pm 1.02}$ / 65.80$_{\pm 2.12}$ |
| | MLP ($D_M$) | 99.43$_{\pm 0.12}$ / 89.90$_{\pm 1.80}$ | 99.74$_{\pm 0.03}$ / 93.40$_{\pm 0.52}$ | 99.18$_{\pm 0.08}$ / 90.18$_{\pm 1.68}$ | 94.86$_{\pm 0.46}$ / 39.00$_{\pm 6.37}$ |
| | Joint ($D_0$) | 99.66$_{\pm 0.10}$ / 91.68$_{\pm 2.41}$ | 99.86$_{\pm 0.06}$ / 96.20$_{\pm 0.25}$ | 99.17$_{\pm 0.28}$ / 90.66$_{\pm 2.76}$ | 96.72$_{\pm 0.91}$ / 54.14$_{\pm 5.04}$ |
| | Attention + MLP ($D_A, D_M$) | 99.74$_{\pm 0.04}$ / 96.40$_{\pm 0.51}$ | 99.92$_{\pm 0.05}$ / 96.74$_{\pm 1.34}$ | 99.68$_{\pm 0.08}$ / 94.92$_{\pm 1.12}$ | 97.30$_{\pm 0.62}$ / 71.32$_{\pm 4.66}$ |
| | **Full** ($D_A, D_M, D_0$) | **99.85**$_{\pm 0.05}$ / **97.16**$_{\pm 1.11}$ | **99.96**$_{\pm 0.03}$ / **99.08**$_{\pm 0.58}$ | **99.78**$_{\pm 0.08}$ / **96.50**$_{\pm 0.53}$ | **98.38**$_{\pm 0.25}$ / **79.68**$_{\pm 2.35}$ |

parisons are consistent with complementary information across the three lesion-response channels, particularly for detection at low FPR.

**Cross-Domain Results.** Table 6 shows a more consistent advantage for combining the three lesion-response channels. The full representation achieves the highest mean AUROC and TPR across all test domains under all four leave-one-domain-out settings. On held-out arXiv, it reaches 98.38% AUROC and 79.68% TPR, exceeding the strongest single-channel results of 96.72% AUROC and 65.80% TPR. On held-out XSum, it achieves 98.13% AUROC and 85.72% TPR, compared with 97.16% and 81.76% for the MLP-only variant. Compared with $(D_A, D_M)$, the full representation achieves higher AUROC and TPR on all four held-out domains, with TPR margins of 4.12–8.36 percentage points. These comparisons further support the additional predictive value of the joint-lesion response beyond the individual attention and MLP responses, particularly under domain shifts.

Table 7: **Detector-branch ablation across LLM sources.** Each cell reports AUROC / TPR@0.1%FPR (mean ± std, %) over five runs. Within each training-source block, the best AUROC and TPR for each test source are independently highlighted in blue and bolded; gray rows denote the dual-branch detector.

| Train | Branches | ChatGPT | Claude-instant | Google-PaLM | Llama-2-70B |
|---|---|---|---|---|---|
| ChatGPT | Sequence only (**g**) | $99.94_{\pm 0.04}$ / $99.74_{\pm 0.09}$ | $95.06_{\pm 0.69}$ / $61.52_{\pm 2.99}$ | $98.10_{\pm 0.63}$ / $85.56_{\pm 0.41}$ | $99.98_{\pm 0.01}$ / $99.34_{\pm 0.23}$ |
| | Statistics only (**u**) | $\mathbf{99.98}_{\pm 0.01}$ / $99.60_{\pm 0.10}$ | $94.78_{\pm 0.83}$ / $57.70_{\pm 6.19}$ | $98.41_{\pm 0.15}$ / $78.34_{\pm 3.56}$ | $99.95_{\pm 0.02}$ / $98.32_{\pm 0.37}$ |
| | **Dual (Ours)** | $99.95_{\pm 0.04}$ / $\mathbf{99.86}_{\pm 0.05}$ | $\mathbf{95.66}_{\pm 0.70}$ / $\mathbf{67.36}_{\pm 2.56}$ | $\mathbf{98.81}_{\pm 0.26}$ / $\mathbf{87.20}_{\pm 1.52}$ | $\mathbf{99.99}_{\pm 0.01}$ / $\mathbf{99.60}_{\pm 0.29}$ |
| Claude | Sequence only (**g**) | $96.49_{\pm 4.01}$ / $46.50_{\pm 31.62}$ | $99.94_{\pm 0.07}$ / $\mathbf{99.44}_{\pm 0.11}$ | $87.88_{\pm 6.33}$ / $26.18_{\pm 16.63}$ | $96.03_{\pm 3.91}$ / $41.02_{\pm 17.83}$ |
| | Statistics only (**u**) | $\mathbf{99.98}_{\pm 0.01}$ / $\mathbf{99.04}_{\pm 0.34}$ | $99.80_{\pm 0.03}$ / $94.84_{\pm 1.89}$ | $\mathbf{98.58}_{\pm 0.60}$ / $\mathbf{79.00}_{\pm 3.30}$ | $\mathbf{99.97}_{\pm 0.00}$ / $\mathbf{97.96}_{\pm 0.70}$ |
| | **Dual (Ours)** | $99.96_{\pm 0.02}$ / $98.38_{\pm 1.14}$ | $\mathbf{99.96}_{\pm 0.05}$ / $99.04_{\pm 1.28}$ | $97.42_{\pm 1.46}$ / $76.88_{\pm 6.61}$ | $99.89_{\pm 0.09}$ / $97.00_{\pm 2.12}$ |
| PaLM | Sequence only (**g**) | $99.96_{\pm 0.03}$ / $99.82_{\pm 0.11}$ | $95.92_{\pm 1.27}$ / $49.86_{\pm 4.62}$ | $99.48_{\pm 0.09}$ / $87.00_{\pm 0.60}$ | $\mathbf{99.99}_{\pm 0.00}$ / $99.68_{\pm 0.16}$ |
| | Statistics only (**u**) | $\mathbf{99.99}_{\pm 0.01}$ / $99.82_{\pm 0.18}$ | $96.95_{\pm 1.04}$ / $63.26_{\pm 4.87}$ | $99.60_{\pm 0.05}$ / $85.64_{\pm 3.72}$ | $\mathbf{99.99}_{\pm 0.00}$ / $98.80_{\pm 0.68}$ |
| | **Dual (Ours)** | $99.97_{\pm 0.02}$ / $\mathbf{99.90}_{\pm 0.00}$ | $\mathbf{97.58}_{\pm 0.27}$ / $\mathbf{68.56}_{\pm 2.58}$ | $\mathbf{99.74}_{\pm 0.04}$ / $\mathbf{92.40}_{\pm 2.17}$ | $\mathbf{99.99}_{\pm 0.00}$ / $\mathbf{99.90}_{\pm 0.07}$ |
| Llama | Sequence only (**g**) | $99.96_{\pm 0.03}$ / $99.76_{\pm 0.11}$ | $95.31_{\pm 0.86}$ / $61.14_{\pm 6.57}$ | $98.10_{\pm 0.52}$ / $85.44_{\pm 1.91}$ | $\mathbf{99.99}_{\pm 0.01}$ / $99.42_{\pm 0.32}$ |
| | Statistics only (**u**) | $\mathbf{99.98}_{\pm 0.01}$ / $99.62_{\pm 0.20}$ | $95.89_{\pm 1.20}$ / $64.80_{\pm 4.53}$ | $98.32_{\pm 0.62}$ / $79.94_{\pm 5.09}$ | $99.96_{\pm 0.02}$ / $98.82_{\pm 0.29}$ |
| | **Dual (Ours)** | $99.96_{\pm 0.03}$ / $\mathbf{99.86}_{\pm 0.09}$ | $\mathbf{96.64}_{\pm 0.79}$ / $\mathbf{69.76}_{\pm 4.54}$ | $\mathbf{98.98}_{\pm 0.26}$ / $\mathbf{89.24}_{\pm 0.93}$ | $\mathbf{99.99}_{\pm 0.00}$ / $\mathbf{99.74}_{\pm 0.05}$ |

Table 8: **Detector-branch ablation across writing domains.** Each cell reports AUROC / TPR@0.1%FPR (mean ± std, %) over five runs. For each leave-one-domain-out setting, the detector is trained on the remaining three domains and evaluated on all four domains. The best AUROC and TPR among the three branch configurations are independently highlighted in blue and bolded; gray rows denote the dual-branch detector.

| Held-out | Branches | Yelp | Writing Prompt | XSum | arXiv |
|---|---|---|---|---|---|
| Yelp | Sequence only (**g**) | $\mathbf{99.42}_{\pm 0.19}$ / $\mathbf{92.50}_{\pm 2.05}$ | $99.92_{\pm 0.07}$ / $97.32_{\pm 1.93}$ | $99.70_{\pm 0.09}$ / $95.22_{\pm 1.73}$ | $99.85_{\pm 0.06}$ / $92.74_{\pm 2.68}$ |
| | Statistics only (**u**) | $98.99_{\pm 0.24}$ / $91.52_{\pm 0.86}$ | $99.95_{\pm 0.03}$ / $97.42_{\pm 0.54}$ | $99.62_{\pm 0.08}$ / $91.16_{\pm 2.57}$ | $99.96_{\pm 0.01}$ / $97.74_{\pm 0.83}$ |
| | **Dual (Ours)** | $99.40_{\pm 0.17}$ / $92.46_{\pm 2.23}$ | $\mathbf{99.96}_{\pm 0.02}$ / $\mathbf{98.62}_{\pm 0.73}$ | $\mathbf{99.79}_{\pm 0.07}$ / $\mathbf{96.00}_{\pm 0.57}$ | $\mathbf{99.97}_{\pm 0.03}$ / $\mathbf{98.20}_{\pm 0.64}$ |
| Writing Prompt | Sequence only (**g**) | $99.77_{\pm 0.07}$ / $96.84_{\pm 0.80}$ | $99.73_{\pm 0.23}$ / $\mathbf{93.76}_{\pm 4.50}$ | $99.69_{\pm 0.10}$ / $95.46_{\pm 2.22}$ | $99.79_{\pm 0.04}$ / $92.40_{\pm 1.05}$ |
| | Statistics only (**u**) | $99.76_{\pm 0.05}$ / $95.50_{\pm 0.37}$ | $99.63_{\pm 0.17}$ / $89.32_{\pm 1.69}$ | $99.53_{\pm 0.08}$ / $91.94_{\pm 0.84}$ | $99.94_{\pm 0.02}$ / $95.64_{\pm 1.91}$ |
| | **Dual (Ours)** | $\mathbf{99.82}_{\pm 0.02}$ / $\mathbf{97.70}_{\pm 0.70}$ | $\mathbf{99.75}_{\pm 0.11}$ / $92.76_{\pm 1.23}$ | $\mathbf{99.83}_{\pm 0.03}$ / $\mathbf{96.40}_{\pm 0.23}$ | $\mathbf{99.97}_{\pm 0.01}$ / $\mathbf{97.26}_{\pm 1.27}$ |
| XSum | Sequence only (**g**) | $99.79_{\pm 0.08}$ / $\mathbf{96.96}_{\pm 1.77}$ | $\mathbf{99.96}_{\pm 0.05}$ / $98.38_{\pm 1.48}$ | $\mathbf{98.25}_{\pm 0.88}$ / $82.76_{\pm 6.93}$ | $99.89_{\pm 0.05}$ / $95.56_{\pm 1.20}$ |
| | Statistics only (**u**) | $99.69_{\pm 0.06}$ / $94.22_{\pm 0.60}$ | $99.95_{\pm 0.01}$ / $97.02_{\pm 0.73}$ | $95.56_{\pm 0.76}$ / $66.34_{\pm 5.19}$ | $99.97_{\pm 0.02}$ / $97.82_{\pm 1.10}$ |
| | **Dual (Ours)** | $\mathbf{99.81}_{\pm 0.03}$ / $96.70_{\pm 1.22}$ | $\mathbf{99.96}_{\pm 0.03}$ / $\mathbf{98.82}_{\pm 0.54}$ | $98.13_{\pm 0.60}$ / $\mathbf{85.72}_{\pm 4.01}$ | $\mathbf{99.99}_{\pm 0.01}$ / $\mathbf{99.40}_{\pm 0.66}$ |
| arXiv | Sequence only (**g**) | $99.79_{\pm 0.07}$ / $96.52_{\pm 1.21}$ | $\mathbf{99.97}_{\pm 0.03}$ / $98.42_{\pm 0.97}$ | $99.70_{\pm 0.04}$ / $94.62_{\pm 2.07}$ | $97.44_{\pm 0.91}$ / $76.84_{\pm 3.50}$ |
| | Statistics only (**u**) | $99.74_{\pm 0.04}$ / $95.72_{\pm 0.48}$ | $99.95_{\pm 0.01}$ / $96.96_{\pm 0.82}$ | $99.65_{\pm 0.03}$ / $93.72_{\pm 0.43}$ | $96.59_{\pm 0.69}$ / $68.70_{\pm 3.17}$ |
| | **Dual (Ours)** | $\mathbf{99.85}_{\pm 0.05}$ / $\mathbf{97.16}_{\pm 1.11}$ | $99.96_{\pm 0.03}$ / $\mathbf{99.08}_{\pm 0.58}$ | $\mathbf{99.78}_{\pm 0.08}$ / $\mathbf{96.50}_{\pm 0.53}$ | $\mathbf{98.38}_{\pm 0.25}$ / $\mathbf{79.68}_{\pm 2.35}$ |

### D.2 Ablation on Detector Branches

**Experimental Setup.** We examine the roles of the two feature-processing branches in our detector. The sequence branch models structured lesion-response patterns across layers and token positions and produces the representation $\mathbf{g}(x)$, whereas the statistics branch summarizes layer-wise distributional characteristics and produces $\mathbf{u}(x)$. We compare a sequence-only variant using $\mathbf{g}(x)$, a statistics-only variant using $\mathbf{u}(x)$, and the full dual-branch detector combining both representations. All variants use the complete three-channel contribution tensor $(D_A, D_M, D_0)$ and share the same proxy model, training objective, data splits, and optimization settings, with only the detector branch structure varied. We follow the same cross-LLM and leave-one-domain-out cross-domain protocols as the main experiments and report mean ± standard deviation over five independent runs.

**Cross-LLM Results.** Table 7 shows that the two branches exhibit different generalization behaviors. The statistics-only detector is particularly competitive when trained on Claude, outperforming both the sequence-only and dual-branch detectors on all three OOD sources. Its AUROC/TPR values reach 99.98%/99.04% on ChatGPT, 98.58%/79.00% on PaLM, and 99.97%/97.96% on Llama-2-70B. The sequence-only detector nevertheless achieves higher TPR than the statistics-only variant in several other settings. Combining the branches yields the highest mean TPR in all 12 settings trained on ChatGPT, PaLM, or Llama. For example, on PaLM-to-Claude transfer, the dual detector achieves 68.56% TPR, compared with 49.86% for sequence-only and 63.26% for statistics-only. Thus, neither individual branch nor the dual detector dominates every setting. The reported results are consistent with complementary contributions from sequential structure and distributional statistics.

**Cross-Domain Results.** Table 8 shows that the sequence-only detector outperforms the statistics-only variant on both metrics across all four held-out domains. It also achieves the highest mean TPR on held-out Yelp and Writing Prompt. The dual detector remains close on Yelp (92.46% versus 92.50%) but trails sequence-only on Writing Prompt (92.76% versus 93.76%). Conversely, combining the branches improves OOD TPR on XSum and arXiv. On XSum, the dual detector achieves 85.72% TPR versus 82.76% for sequence-only, although its AUROC is slightly lower (98.13% versus 98.25%). On arXiv, it achieves 98.38% AUROC and 79.68% TPR, exceeding sequence-only at 97.44%/76.84% and statistics-only at 96.59%/68.70%. These results highlight the sequence branch's effectiveness under domain shifts, while showing that the statistics branch can further improve the combined detector. Together with the cross-LLM results, they support the value of combining the two representations without implying uniform gains across every transfer setting.

### D.3 Comparison of Classification Losses

**Experimental Setup.** We examine the effect of the classification loss by comparing the focal loss used in our detector with standard binary cross-entropy (BCE). All other components, including the contribution features, detector architecture, data splits, and optimization settings, are kept unchanged. We follow the same cross-LLM and leave-one-domain-out cross-domain protocols as the main experiments and report mean performance over five independent runs. To highlight the effect of the loss function, we visualize the performance difference $\Delta = \text{Focal} - \text{BCE}$, where positive values indicate improvements from focal loss.

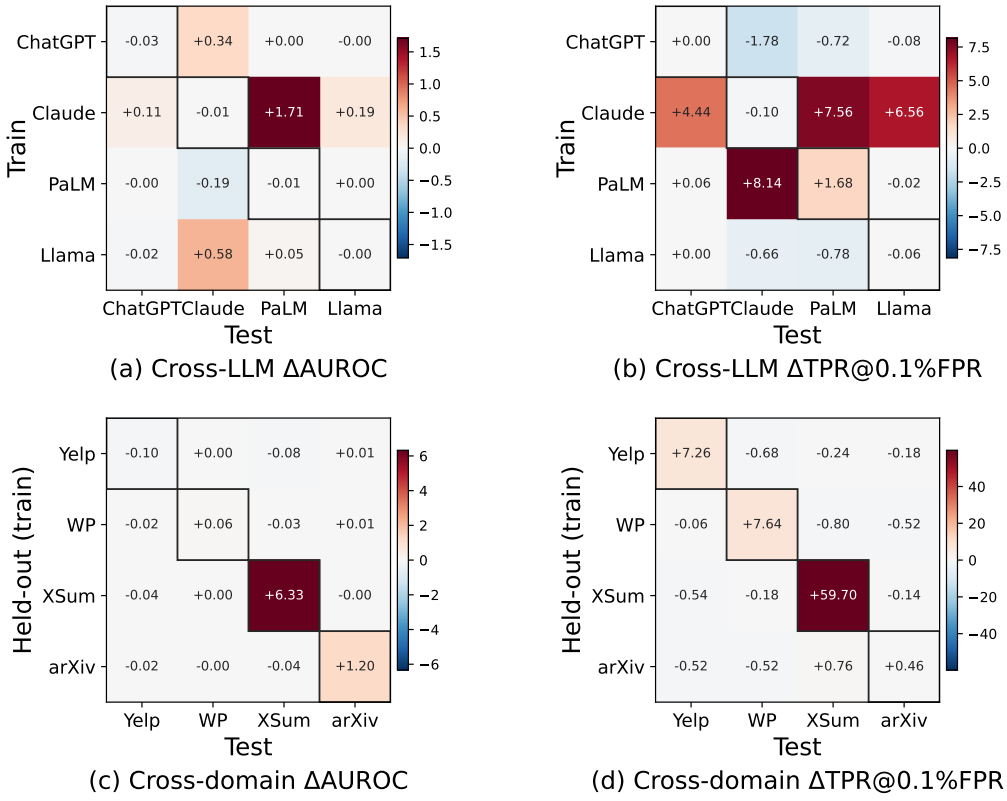


Figure 7: **Effect of focal loss relative to BCE.** Each heatmap reports the performance difference $\Delta = \text{Focal} - \text{BCE}$ in percentage points. (a) Cross-LLM ΔAUROC. (b) Cross-LLM ΔTPR@0.1%FPR. (c) Cross-domain ΔAUROC. (d) Cross-domain ΔTPR@0.1%FPR. In (a) and (b), rows and columns denote training and test LLM sources, respectively; diagonal cells correspond to ID evaluation and off-diagonal cells to cross-source OOD evaluation. In (c) and (d), rows denote the held-out training domain and columns denote the test domain, such that diagonal cells correspond to evaluation on the held-out OOD domain. Positive values indicate improvements from focal loss, while negative values favor BCE.

**Cross-LLM Results.** Figures 7(a)–(b) show that focal loss and BCE perform similarly in many ID or already saturated settings, whereas larger differences emerge under particular cross-source shifts. The most notable TPR improvements occur when training on Claude and testing on ChatGPT, PaLM, and Llama-2-70B, with gains of $+4.44$, $+7.56$, and $+6.56$ percentage points, respectively. Focal loss also improves TPR by $+8.14$ points when training on PaLM and testing on Claude. The corresponding AUROC changes are generally smaller, although Claude→PaLM improves by $+1.71$ points. These results indicate that focal loss primarily benefits more difficult source-transfer settings rather than uniformly improving already well-separated cases.

**Cross-Domain Results.** Figures 7(c)–(d) reveal a clearer pattern on the held-out domains. Along the diagonal, focal loss improves TPR by $+7.26$, $+7.64$, $+59.70$, and $+0.46$ percentage points on Yelp, Writing Prompt, XSum, and arXiv, respectively. The largest gain occurs on held-out XSum, where focal loss improves AUROC by $+6.33$ points and TPR by $+59.70$ points. In contrast, most non-diagonal domain evaluations show only small changes around zero. This suggests that the advantage of focal loss is concentrated on harder OOD examples, particularly under the stringent low-FPR operating point.

Overall, the results support focal loss as a more robust classification objective under challenging distribution shifts, while its effect remains limited when BCE already achieves near-saturated performance.

## E Sensitivity to Test-Set Size

We assess the sensitivity of LesionProbe's performance estimates to test-set size under mixed-source ID evaluation, cross-LLM OOD transfer, and cross-domain OOD transfer, with particular attention to TPR@0.1%FPR.

**Experimental Setup.** We construct separate test pools for mixed-source ID evaluation, cross-LLM OOD transfer, and cross-domain OOD transfer, keeping training and test data disjoint throughout. For ID evaluation, we construct a balanced mixed-source test pool of 10,000 matched HWT–LGT pairs. For cross-LLM OOD evaluation, each detector is trained on one LLM source and evaluated on 8,400 pairs pooled from the other three sources, with 2,800 pairs per source. For cross-domain OOD evaluation, each detector is trained on one domain and evaluated on 8,400 pairs pooled from the other three domains, with 2,800 pairs per domain. The maximum test sizes reflect the available samples under these respective splits. Pooling the unseen sources or domains enables larger-scale evaluation than individual subsets permit, while excluding the training source or domain from the OOD test pool.

For each training setting, we reuse five detectors trained with different random seeds, each using 512 matched HWT–LGT training pairs. The ID detectors use mixed-source training data, whereas the OOD detectors use the corresponding single-source or single-domain training data. All detectors remain fixed, without retraining or hyperparameter adjustment. Within each test pool, we construct nested subsets of 1,000, 3,000, and 5,000 pairs using random seed 42 and also evaluate the full pool. Sampling is stratified jointly by generation source and writing domain at the pair level, preserving HWT–LGT pairings. All five detectors within a setting share the same test subsets. We compute AUROC and TPR@0.1%FPR using the same empirical ROC interpolation procedure across all three evaluation settings. For OOD evaluation, metrics are computed on the pooled test examples rather than averaged across separately evaluated sources or domains.

**Uncertainty Estimation.** We report the mean and standard deviation across the five detectors to characterize variation across training runs. We separately quantify test-sampling uncertainty using 1,000 stratified paired-bootstrap replicates at each test size. Pairs are sampled with replacement within each stratum, preserving stratum counts, and the same resampled examples are used for all five detectors. For each replicate, we recompute each detector's metrics and then average them across detectors. The 2.5th and 97.5th percentiles of these bootstrap means define approximate 95% confidence intervals. These intervals characterize test-sampling uncertainty conditional on the five fixed detectors, rather than variation across training runs.

**ID Results.** Table 9 shows little variation in the point estimates across the four test sizes. Mean AUROC remains between 99.85% and 99.86%, while mean TPR ranges from 94.49% to 95.40%. From 1,000 to 10,000 pairs, AUROC is unchanged at the reported precision, and TPR decreases by

Table 9: **Sensitivity to test-set size under mixed-source ID evaluation.** Each pair contains one HWT and one LGT sample. Mean ± SD summarizes five fixed models. Confidence intervals are approximate 95% percentile intervals for the five-model mean, obtained from 1,000 stratified paired-bootstrap replicates. The test subsets are nested, and all metric values are percentages.

| | AUROC | | TPR@0.1%FPR | |
|---|---|---|---|---|
| **Test pairs** | **Mean ± SD** | **95% CI** | **Mean ± SD** | **95% CI** |
| 1,000 | $99.86 \pm 0.03$ | [99.79, 99.93] | $95.40 \pm 0.82$ | [93.44, 97.10] |
| 3,000 | $99.85 \pm 0.04$ | [99.79, 99.89] | $94.49 \pm 0.93$ | [93.10, 95.96] |
| 5,000 | $99.85 \pm 0.03$ | [99.81, 99.88] | $94.66 \pm 0.81$ | [93.51, 95.58] |
| 10,000 | $99.86 \pm 0.03$ | [99.83, 99.89] | $95.13 \pm 0.67$ | [94.43, 95.65] |

Table 10: **Sensitivity to test-set size under cross-LLM OOD evaluation.** For each training source, evaluation pools examples from the other three LLM sources. Mean ± SD and 95% confidence intervals follow the definitions in Table 9. Subsets are nested within each setting, and all metric values are percentages.

| | | AUROC | | TPR@0.1%FPR | |
|---|---|---|---|---|---|
| **Training source** | **Test pairs** | **Mean ± SD** | **95% CI** | **Mean ± SD** | **95% CI** |
| ChatGPT | 1,000 | $98.26 \pm 0.25$ | [97.87, 98.66] | $81.02 \pm 2.32$ | [77.72, 84.50] |
| | 3,000 | $98.16 \pm 0.25$ | [97.88, 98.42] | $81.63 \pm 2.28$ | [79.48, 83.43] |
| | 5,000 | $98.08 \pm 0.25$ | [97.86, 98.28] | $80.14 \pm 2.77$ | [78.46, 81.93] |
| | 8,400 | $98.08 \pm 0.24$ | [97.91, 98.24] | $80.59 \pm 2.11$ | [79.31, 81.80] |
| Claude-instant | 1,000 | $98.44 \pm 0.85$ | [98.11, 98.79] | $80.34 \pm 12.75$ | [62.76, 85.50] |
| | 3,000 | $98.32 \pm 0.83$ | [98.08, 98.54] | $78.64 \pm 11.50$ | [66.40, 83.07] |
| | 5,000 | $98.38 \pm 0.77$ | [98.20, 98.55] | $78.85 \pm 10.97$ | [73.26, 82.36] |
| | 8,400 | $98.40 \pm 0.78$ | [98.26, 98.53] | $79.15 \pm 11.50$ | [75.39, 81.09] |
| Google-PaLM | 1,000 | $99.06 \pm 0.09$ | [98.75, 99.34] | $81.82 \pm 2.62$ | [79.96, 88.86] |
| | 3,000 | $98.89 \pm 0.13$ | [98.69, 99.05] | $81.51 \pm 2.08$ | [79.50, 84.19] |
| | 5,000 | $98.87 \pm 0.13$ | [98.73, 99.02] | $81.70 \pm 2.33$ | [80.26, 83.86] |
| | 8,400 | $98.89 \pm 0.11$ | [98.78, 99.00] | $81.48 \pm 2.20$ | [80.21, 83.04] |
| Llama-2-70B | 1,000 | $98.22 \pm 0.48$ | [97.73, 98.66] | $83.50 \pm 2.70$ | [80.90, 86.50] |
| | 3,000 | $98.14 \pm 0.57$ | [97.86, 98.39] | $82.01 \pm 1.47$ | [79.23, 84.42] |
| | 5,000 | $98.03 \pm 0.60$ | [97.82, 98.24] | $82.15 \pm 0.96$ | [80.17, 83.81] |
| | 8,400 | $98.03 \pm 0.59$ | [97.88, 98.19] | $81.75 \pm 1.25$ | [80.46, 83.08] |

0.27 percentage points. Meanwhile, the TPR confidence interval narrows from $[93.44\%, 97.10\%]$ to $[94.43\%, 95.65\%]$. Its width decreases from 3.66 to 1.22 percentage points, indicating improved estimation precision with the larger test set.

**Cross-LLM OOD Results.** Table 10 shows similarly limited sensitivity of the mean estimates to test-set size under cross-LLM transfer. Within each training-source setting, AUROC varies by at most 0.19 percentage points across the four sizes, and TPR varies by at most 1.75 points. From 1,000 to 8,400 pairs, mean TPR decreases by 0.34–1.75 points across the four settings. The bootstrap confidence intervals also narrow with increasing test size. For ChatGPT-trained models, for example, the TPR interval contracts from $[77.72\%, 84.50\%]$ to $[79.31\%, 81.80\%]$. However, limited sensitivity to test size does not imply uniformly low variation across training runs. Claude-instant-trained models retain a TPR standard deviation of 11.50 points at 8,400 pairs, compared with 1.25–2.20 points for the other training sources.

**Cross-Domain OOD Results.** Table 11 shows that the smaller test subsets broadly preserve the performance estimates obtained from larger unseen-domain pools. Within each training-domain setting, AUROC varies by at most 0.33 percentage points across the four test sizes, while TPR varies by at most 2.78 points. From 1,000 to 8,400 pairs, TPR changes by +2.27, -0.62, -2.78,

Table 11: **Sensitivity to test-set size under cross-domain OOD evaluation.** For each training domain, evaluation pools examples from the other three domains. Mean ± SD and 95% confidence intervals follow the definitions in Table 9. Subsets are nested within each setting, and all metric values are percentages.

| | | AUROC | | TPR@0.1%FPR | |
|---|---|---|---|---|---|
| **Training domain** | **Test pairs** | **Mean ± SD** | **95% CI** | **Mean ± SD** | **95% CI** |
| arXiv | 1,000 | $90.19 \pm 3.12$ | [89.17, 91.22] | $26.10 \pm 5.63$ | [23.64, 32.44] |
| | 3,000 | $90.14 \pm 3.08$ | [89.54, 90.74] | $26.44 \pm 6.07$ | [23.72, 29.97] |
| | 5,000 | $90.09 \pm 3.05$ | [89.65, 90.54] | $28.77 \pm 7.55$ | [26.18, 30.79] |
| | 8,400 | $90.25 \pm 3.07$ | [89.93, 90.59] | $28.37 \pm 7.05$ | [26.25, 30.16] |
| Writing Prompt | 1,000 | $88.99 \pm 1.35$ | [88.00, 89.89] | $48.30 \pm 4.68$ | [45.84, 53.32] |
| | 3,000 | $89.32 \pm 1.19$ | [88.74, 89.80] | $48.61 \pm 4.17$ | [46.11, 51.33] |
| | 5,000 | $89.26 \pm 1.22$ | [88.81, 89.67] | $47.49 \pm 4.08$ | [45.30, 49.36] |
| | 8,400 | $89.14 \pm 1.23$ | [88.81, 89.45] | $47.68 \pm 4.06$ | [46.07, 49.23] |
| XSum | 1,000 | $98.56 \pm 0.41$ | [98.14, 98.92] | $86.62 \pm 2.77$ | [84.46, 88.90] |
| | 3,000 | $98.50 \pm 0.43$ | [98.29, 98.70] | $84.11 \pm 3.23$ | [81.85, 85.80] |
| | 5,000 | $98.51 \pm 0.45$ | [98.33, 98.67] | $84.12 \pm 3.26$ | [82.34, 85.33] |
| | 8,400 | $98.47 \pm 0.45$ | [98.34, 98.60] | $83.84 \pm 3.22$ | [82.16, 84.91] |
| Yelp | 1,000 | $97.37 \pm 0.51$ | [96.89, 97.86] | $71.74 \pm 7.88$ | [67.42, 77.12] |
| | 3,000 | $97.39 \pm 0.58$ | [97.12, 97.64] | $72.12 \pm 6.32$ | [69.69, 76.62] |
| | 5,000 | $97.38 \pm 0.52$ | [97.16, 97.60] | $72.93 \pm 3.94$ | [70.29, 75.48] |
| | 8,400 | $97.25 \pm 0.53$ | [97.08, 97.41] | $72.65 \pm 3.89$ | [70.13, 74.17] |

and +0.91 points for arXiv-, Writing Prompt-, XSum-, and Yelp-trained models, respectively. The changes occur in both directions, and the ordering of training-domain settings by mean TPR remains unchanged across test sizes. Thus, enlarging the test pool does not reveal a consistent upward bias in the smaller-subset estimates or alter the broad pattern of transfer performance. Meanwhile, the bootstrap confidence intervals narrow in every setting. For arXiv-trained models, for example, the TPR interval contracts from $[23.64\%, 32.44\%]$ to $[26.25\%, 30.16\%]$. These results indicate that larger test pools primarily improve estimation precision, while the 1,000-pair subsets retain broadly comparable point estimates.

**Interpolation Check and Overall Findings.** We additionally compute the maximum empirical TPR subject to $\mathrm{FPR} \leq 0.001$ without interpolation on each full test pool. The resulting five-model means match the interpolated means at the reported precision for ID evaluation and all eight OOD settings. This agreement shows that the reported full-pool means are not materially changed by interpolation at that precision. Comparing 1,000-pair subsets with the full pools, the absolute differences in mean TPR are 0.27 percentage points under ID evaluation, at most 1.75 points under cross-LLM transfer, and at most 2.78 points under cross-domain transfer. Together, these results provide complementary evidence that the smaller-test-set estimates broadly reflect performance on larger pools across the evaluated ID and OOD settings. The bootstrap intervals further quantify test-sampling uncertainty that is not captured by variation across training runs. Given the available benchmark sizes, pooling unseen sources or domains enables this larger-scale sensitivity check while preserving separation from the training source or domain. These findings strengthen confidence in the empirical TPR@0.1%FPR estimates, without establishing the uncertainty of individual method differences or guaranteeing a 0.1% FPR for a fixed deployment threshold.